

Université Paris VII
Centre d'études et de recherches
en informatique linguistique

RECONNAISSANCE DE SEQUENCES DE MOTS
PAR AUTOMATE

Adverbes de date du Français

Denis MAUREL

Volume II
Programmes et fichiers

Thèse de doctorat en informatique fondamentale
Directeur de thèse: Maurice Gross
Octobre 1989



Jury:

Dominique Perrin (*Président*)
Maxime Crochemore
Laurence Danlos
Maurice Gross
Maurice Salkoff



1989
NAU

Université Paris VII
Centre d'études et de recherches
en informatique linguistique

**RECONNAISSANCE DE SEQUENCES DE MOTS
PAR AUTOMATE**

Adverbes de date du Français

Denis MAUREL

Volume II
Programmes et fichiers

Thèse de doctorat en informatique fondamentale
Directeur de thèse: Maurice Gross
Octobre 1989

Jury:

Dominique Perrin (*Président*)
Maxime Crochemore
Laurence Danlos
Maurice Gross
Maurice Salkoff

I

9. PROGRAMMES.

9.1. FICHIER *ADVDATE.C*

```
# include "definir.h"

int finarti=0,          /* teste la fin de l'article */
    sort_imp=0,         /* sortie sur imprimante */
    sort_res=1;        /* sortie sur le fichier des résidus */
FICHIER *fic_indx,      /* fichier index */
        *fic_mots,      /* fichier dictionnaire */
        *fic_tran,      /* fichier automate */
        *fic_res;       /* second fichier des résidus */
MOTS *module1=RIEN,     /* état du module 1 */
    *module2=RIEN,      /* état du module 2 */
    *module3=RIEN;      /* état du module 3 */

void main(argc, argv)

int argc;
char *argv[];           /* nom du fichier à lire */
{
    FICHIER *article;    /* fichier à étudier */
    char nomfich[MAXFICH+1]; /* noms des fichiers */
    MOTS *pre_mots, *cou_mots;
    extern char *menu(), *chargnom();
    extern void titre(), ecran(), debut(),
                imprime(), fermerur(), fin_impr();
    extern FICHIER *ouvertur();
    extern MOTS *cl_mots(), *dnum(), *compauto(), *tabtrans();

    titre();
    if (argc-1) cop_ch(nomfich, argv[1]);
    else cop_ch(nomfich, menu());

        /****** ouverture des fichiers *****/

    article=ouvertur(nomfich, 1);

    cop_ch(nomfich, chargnom(1));
    fic_indx=ouvertur(nomfich, 1);
    cop_ch(nomfich, chargnom(2));
    fic_mots=ouvertur(nomfich, 1);
    cop_ch(nomfich, chargnom(3));
    fic_tran=ouvertur(nomfich, 1);
    cop_ch(nomfich, chargnom(4));
    fic_res=ouvertur(nomfich, 0);

        /****** module de consultation du dictionnaire *****/

    debut(article);
    do
    {
```

```

if ((!module2)&&(!module3))
{
    pre_mots=cl_mots(article);
    pre_mots=dnum(pre_mots, pre_mots);
}

/***** module de consultation de l'automate *****/

if (!module3) pre_mots=compauto(pre_mots);

/***** module de consultation des tables de transitions *****/

pre_mots=tabtrans(pre_mots);

/***** mots reconnus *****/

cou_mots=pre_mots;
while(cou_mots)
{
    ecran(cou_mots->mot, 'I');
    ecran(cou_mots->car, 'I');
    if (sort_imp)
    {
        imprime(cou_mots->mot, 'S');
        imprime(cou_mots->car, 'S');
    }
    libere(cou_mots->motdico);
    libere(cou_mots);
    cou_mots=cou_mots->suivant;
}
}
while ((!finarti) || (module1) || (module2) || (module3));

/***** fermeture des fichiers *****/

fermetur(article);
fermetur(fic_indx);
fermetur(fic_mots);
fermetur(fic_tran);
fermetur(fic_res);
if (sort_imp) fin_impr();
}

```

9.2. FICHIER *MODULE0.C*

```

#include "definir.h"

extern FICHIER *fic_indx,          /* fichier index */
               *fic_mots;         /* fichier dictionnaire */

/* crée l'index du dictionnaire en groupant ensemble */
/* les expressions composés commençant par le même mot et */
/* les mots ne différant que par des lettres accentuées */

void creeindx()
{
    char nomfich[MAXFICH],          /* nom de fichier */
          mot[MAXMOT+2];            /* mot lu */
    int i,                          /* indentation */
        saut=0,                    /* nombre de caractères avant le mot */
        nbcar=0,                   /* nombre de caractères lus sur une série */
        nb,                        /* nombre de caractères lus sur un terme */
        reduc=0;                   /* si réduction de la liste */
    INDEX *pre_indx, *cou_indx, *der_indx,
           *rep_indx;              /* reprise */
    extern char *lit_mot(), *chargnom();
    extern lit_nbca(), alpha();
    extern INDEX *cp_indx();
    extern FICHIER *ouvertur();
    extern void fermetur();

    cop_ch(nomfich, chargnom(1));
    fic_indx=ouvertur(nomfich, -1);
    cop_ch(nomfich, chargnom(2));
    fic_mots=ouvertur(nomfich, 1);

    /* création d'une liste linéaire chaînée */
    /* de LIGNINDX octets */

    /* premier pointeur */

    cop_ch(mot, lit_mot(fic_mots));
    nb=1_ch(mot)+1+lit_nbca(mot);
    pre_indx=cp_indx(mot, saut);
    nbcar+=nb;
    saut+=nb;

    /* pointeurs suivants */

    cou_indx=pre_indx;
    cop_ch(mot, lit_mot(fic_mots));
    nb=1_ch(mot)+1+lit_nbca(mot);
    nbcar+=nb;
    while (mot[0])
    {
        i=1;
        do
        {
            while ((mot[0])&&(nbcar<=i*LIGNINDX))
            {

```

```

        der_indx=cp_indx(mot, saut);
        saut+=nb;
        cou_indx->suivant=der_indx;
        cou_indx=der_indx;
        cop_ch(mot, lit_mot(fic_mots));
        nb=l_ch(mot)+1+lit_nbca(mot);
        nbcar+=nb;
    }
    i+=1;
}
while (!alpha(mot, pre_indx->mot));
if (mot[0])
{
    rep_indx=cp_indx(mot, saut);
    saut+=nb;
    der_indx->suivant=rep_indx;

    /* rétrécissement de la liste */

    if (!alpha(rep_indx->mot, der_indx->mot))
    {
        reduc=1;
        cou_indx=pre_indx;
        while (alpha(cou_indx->mot, der_indx->mot))
            cou_indx=cou_indx->suivant;
        rep_indx=cou_indx;
    }

    /* enregistrement dans l'index */

    if ((fprintf(fic_indx, "%s%d|", pre_indx->mot, pre_indx-
> saut))<0) printf("\nAttention ! %s%d| n'a pas été écrit !", pre_indx->mot, pre_indx-
> saut);

    /* effacement des termes enregistrés */

    cou_indx=pre_indx;
    do libere(cou_indx);
    while ((cou_indx=cou_indx->suivant)!=rep_indx);

    /* nouvelle liste */

    pre_indx=rep_indx;
    if (!reduc)
    {
        cou_indx=pre_indx;
        nbcar=0;
    }
    else
    {
        reduc=0;
        cou_indx=der_indx->suivant;
        nbcar=(cou_indx->saut)-(pre_indx->saut)+nb;
    }
    cop_ch(mot, lit_mot(fic_mots));
    nb=l_ch(mot)+1+lit_nbca(mot);
}

```

```
}

/* enregistrement dans l'index */
/* des derniers termes */

if ((fprintf(fic_indx, "%s%d|", pre_indx->mot, pre_indx->saut)) < 0)
printf("\nAttention ! '%s%d|' n'a pas été écrit !", pre_indx->mot, pre_indx->saut);

cou_indx = pre_indx;
do libere(cou_indx);
while ((cou_indx = cou_indx->suivant));

fermetur(fic_indx);
fermetur(fic_mots);
}
```

9.3. FICHER MODULE1.C

```

#include "definir.h"
#define MAXCHAIN 50

extern finarti,          /* teste la fin du fichier */
      sort_imp ;        /* sortie sur imprimante */
extern MOTS *module1 ;   /* état du module1 */

MOTS *cl_mots(article)

FICHER *article ;        /* fichier à lire */
{
    char terme[MAXMOT+MAXPONCT+1], /* terme lu */
          mot[MAXMOT+1],           /* mot lu */
          car[MAXPONCT],          /* caractères suivant le mot lu */
          format,                 /* pour min_maj */
          c ;                     /* caractère lu */
    int i, j, k,                  /* indentation */
        complemt=MAXPART,         /* de la liste */
        longueur,                 /* d'une chaîne de caractères */
        motsimpl,                 /* compte les mots simple pour un mot */
        decalage,                 /* compte le décalage pour un mot */
        finliste=0 ;              /* teste la fin de la liste */
    MOTS *pre_mots, *der_mots, *cou_mots,
          *ici_mots ;              /* endroit de l'étude */
    DICO *pre_dico, *cou_dico, *sup_dico, *lir_dico ;
    extern char *lit_term(), *test_car(), *cas_ltch(),
               *min_maj() ;
    extern est_ltnb(), mot_comp(), fin_adv() ;
    extern void ecrire(), imprime() ;
    extern DICO *cp_dico(), *chrchdic() ;
    extern MOTS *cp_mots() ;

    if (!module1)
    {
        cop_ch(terme, lit_term(article)) ;
        if (!terme[0])
        {
            finarti=1 ;
            return(RIEN) ;
        }
        longueur=l_ch(terme) ;
        for (i=0 ; est_ltnb(terme[i]) ; i++)
            mot[i]=terme[i] ;
        mot[i]='\0' ;
        car[0]=terme[i++] ;
        for (j=1 ; i<longueur ; i++, j++) car[j]=terme[i] ;
        car[j]='\0' ;
        pre_mots=cp_mots(mot, car) ;
        pre_mots->motdico=cp_dico(cas_ltch(pre_mots, test_car(pre_mots-
>mot)), "", "");
        cou_mots=pre_mots ;
        ici_mots=pre_mots ;
        finliste=fin_adv(cou_mots->car) ;
        if (finliste) for (i=0 ; i<MAXPART ; i++)
        {
            der_mots=cp_mots("", "") ;

```

```

        der_mots->precedent=cou_mots ;
        cou_mots->suivant=der_mots ;
        cou_mots=der_mots ;
    }
}
else /* le premier MOTS conservé n'est pas reconnu */
{
    ecrire(module1->mot) ;
    ecrire(module1->car) ;
    if (sort_imp)
    {
        imprime(module1->mot, 'C') ;
        imprime(module1->car, 'C') ;
    }
    libere(module1) ;
    if (!module1->suivant)
    {
        module1=RIEN ;
        return(cl_mots(article)) ;
    }
    pre_mots=module1->suivant ;
    pre_mots->precedent=RIEN ;
    cou_mots=pre_mots ;
    ici_mots=pre_mots ;
    while (cou_mots->suivant)
    {
        complemt-- ;
        cou_mots=cou_mots->suivant ;
    }
    der_mots=cou_mots ;
    finliste=fin_adv(cou_mots->car) ;
    if ((finliste) || (finarti)) for (i=0 ; i<MAXPART ; i++)
    {
        der_mots=cp_mots("", "") ;
        der_mots->precedent=cou_mots ;
        cou_mots->suivant=der_mots ;
        cou_mots=der_mots ;
    }
}

/* pointeurs suivants (MAXPART) */
if ((!finliste)&&(!finarti)) for (k=0 ; k<complemt ; k++)
{
    cop_ch(terme, lit_term(article)) ;
    if (!terme[0])
    {
        finarti=1 ;
        for (i=0 ; i<MAXPART ; i++)
        {
            der_mots=cp_mots("", "") ;
            der_mots->precedent=cou_mots ;
            cou_mots->suivant=der_mots ;
            cou_mots=der_mots ;
        }
        break ;
    }
}
longueur=l_ch(terme) ;

```

```

for (i=0 ; est_ltnb(terme[i]) ; i++)
    mot[i]=terme[i] ;
mot[i]='\0' ;
car[0]=terme[i++] ;
for (j=1 ; i<longueur ; i++, j++)
    car[j]=terme[i] ;
car[j]='\0' ;
der_mots=cp_mots(mot, car) ;
der_mots->motdico=cp_dico(cas_lrch(der_mots, test_car(der_mots-
>mot)), "", "") ;
der_mots->precedent=cou_mots ;
cou_mots->suivant=der_mots ;
cou_mots=der_mots ;
finliste=fin_adv(cou_mots->car) ;
if (finliste)
{
    for (i=0 ; i<MAXPART ; i++)
    {
        der_mots=cp_mots("", "") ;
        der_mots->precedent=cou_mots ;
        cou_mots->suivant=der_mots ;
        cou_mots=der_mots ;
    }
    break ;
}
}

/* Etude du terme ici */

while (ici_mots->mot[0])
{
    /* recherche dans le dictionnaire */

    lir_dico=ici_mots->motdico ;
    pre_dico=chrchdic(lir_dico->mot) ;

    cou_dico=pre_dico ;
    decalage=0 ;
    while (cou_dico)
    {
        motsimpl=mot_comp(ici_mots, cou_dico->mot) ;
        if (motsimpl>decalage)
        {
            sup_dico=cou_dico ;
            decalage=motsimpl ;
        }
        cou_dico=cou_dico->suivant ;
    }
    if (decalage>0) /* mot trouvé */
    {
        format=test_car(lir_dico->mot)[0] ;
        cop_ch(lir_dico->mot, sup_dico->mot) ;
        cop_ch(lir_dico->gram, sup_dico->gram) ;
        cop_ch(lir_dico->temps, sup_dico->temps) ;
        if (sup_dico->suivant)
            cop_ch(mot, (sup_dico->suivant)->mot) ;
    }
}

```

```

while ((sup_dico->suivant)&&
(!alphacct(min_maj(sup_dico->mot, format), min_maj((sup_dico->suivant)->mot,
format))))
{
    sup_dico=sup_dico->suivant ;
    lir_dico->suivant=cp_dico(mot, sup_dico->gram, sup_dico-
>temps) ;
    lir_dico=lir_dico->suivant ;
    if (sup_dico->suivant)
        cop_ch(mot, (sup_dico->suivant)->mot);
}
cou_dico=pre_dico ;
do libere(cou_dico) ;
while ((cou_dico=cou_dico->suivant)) ;

/* Décalage de la suite linéaire chaînée */

cou_mots=ici_mots ;
for (k=1 ; (k==1) || (k<=decalage) ; k++)
    cou_mots=cou_mots->suivant ;
ici_mots->decale=cou_mots ;

/* libération des DICO au milieu du décalage */

cou_mots=ici_mots->suivant ;
while (cou_mots!=ici_mots->decale)
{
    cou_dico=cou_mots->motdico ;
    libere(cou_dico) ;
    cou_mots->motdico=RIEN ;
    cou_mots=cou_mots->suivant ;
}

/* ajout de termes supplémentaires */

if ((!finliste)&&(!finarti))
    for(k=0 ; (!k) || (k<decalage) ; k++)
    {
        cou_mots=der_mots ;
        cop_ch(terme, lit_term(article)) ;
        if (!terme[0])
        {
            for (i=1 ; i<=MAXPART ; i++)
            {
                der_mots=cp_mots("", "") ;
                der_mots->precedent=cou_mots ;
                cou_mots->suivant=der_mots ;
                cou_mots=der_mots ;
            }
            finarti=1 ;
            break ;
        }
        longueur=l_ch(terme) ;
        for (i=0 ; (est_ltnb(terme[i])) ; i++)
            mot[i]=terme[i] ;
        mot[i]='\0' ;
        car[0]=terme[i++] ;
    }

```

```

    for (j=1 ; i<longueur ; i++ , j++)
        car[j]=terme[i] ;
    car[j]='\0' ;
    der_mots=cp_mots(mot, car) ;
    der_mots->motdico=cp_dico(cas_lrch(der_mots,
test_car(der_mots->mot)), "", "");
    der_mots->precedent=cou_mots ;
    cou_mots->suivant=der_mots ;
    cou_mots=der_mots ;
    finliste=fin_adv(der_mots->car) ;
    if (finliste)
    {
        for (i=0 ; i<MAXPART ; i++)
        {
            der_mots=cp_mots("", "");
            der_mots->precedent=cou_mots ;
            cou_mots->suivant=der_mots ;
            cou_mots=der_mots ;
        }
        break ;
    }
}

/* suppression des premiers termes */

if (!decalage)
{
    if (ici_mots==pre_mots)
        /* (rien avant le mot à supprimer) */
    {
        cou_dico=pre_mots->motdico ;
        libere(cou_dico) ;
        ecrire(pre_mots->mot) ;
        ecrire(pre_mots->car) ;
        if (sort_imp)
        {
            imprime(pre_mots->mot, 'C') ;
            imprime(pre_mots->car, 'C') ;
        }
        libere(pre_mots) ;
        pre_mots=ici_mots->suivant ;
        pre_mots->precedent=RIEN ;
    }
    else
        /* (plusieurs mots avant celui à supprimer) */
    {
        cou_mots=ici_mots->precedent ;
        cou_mots->suivant=RIEN ;
        while (!cou_mots->motdico)
            cou_mots=cou_mots->precedent ;
        cou_mots->decale=RIEN ;
        ici_mots->precedent=RIEN ;
        cou_dico=ici_mots->motdico ;
        libere(cou_dico) ;
        module1=ici_mots ;
        cou_mots=ici_mots ;
        while ((cou_mots->suivant)->mot[0])
            cou_mots=cou_mots->suivant ;
        der_mots=cou_mots ;
    }
}

```

```

        while ((cou_mots=cou_mots->suivant))
            libere(cou_mots);
        der_mots->suivant=RIEN;
        cou_mots=der_mots;
        while (!cou_mots->motdico)
            cou_mots=cou_mots->precedent;
        cou_mots->decale=RIEN;

        return(pre_mots);
    }
}
ici_mots=ici_mots->decale;
}

/* fin de liste */

if (ici_mots!=pre_mots)
{
    cou_mots=ici_mots->precedent;
    cou_mots->suivant=RIEN;
    while (!cou_mots->motdico)
        cou_mots=cou_mots->precedent;
    cou_mots->decale=RIEN;
}
module1=RIEN;
cou_mots=ici_mots;
do libere(cou_mots);
while ((cou_mots=cou_mots->suivant));
if (!decalage) /* rien n'a été reconnu */
    return(cl_mots(article));
return(pre_mots);
}

```

9.4. FICHIER *MODULE1A.C*

```

# include "definir.h"

/* contraction des déterminants numériques */
/* écrits en toutes lettres */

MOTS *dnum(pre_mots, ptr_mots)

MOTS *pre_mots, /* de la liste */
    *ptr_mots ; /* début de l'étude */
{
    int test, /* teste les Dnum */
        testieme=0, /* teste les Dnumièmes */
        tableau[MAXDNUM], /* des valeurs chiffrées */
        dim=2, /* dimension */
        i, j, k ; /* indentation */
    MOTS *cou_mots,
        *dn_mots=RIEN, /* premier dnum */
        *um_mots=RIEN ; /* dernier dnum */
    DICO *cou_dico ;
    extern void écrire() ;
    extern alpha() ;
    extern DICO *cp_dico() ;

/* tests */

    if (!pre_mots) return(RIEN) ;
    cou_mots=ptr_mots ;
    while (cou_mots)
    {
        cou_dico=cou_mots->motdico ;
        test=0 ;
        while (cou_dico)
        {
            /* recherche du premier Dnum */

            if (!comp_ch(cou_dico->temps,"DN"))
            {
                dn_mots=cou_mots ;
                test=1 ;
                break ;
            }
            cou_dico=cou_dico->suivant ;
        }
        if (dn_mots)
        {
            /* élimination des chiffres arabes ou romains */
            /* ou des nombres non suivis d'un blanc ou d'un tiret */

            if (((cou_dico->mot[0]='d')&&
                (cou_dico->mot[1]='n')) ||
                ((cou_dico->mot[0]='r')&&(cou_dico->mot[1]='m')) ||
                ((cou_mots->car[0]!=' ')&&(cou_mots->car[0]!='-'))))
                dn_mots=RIEN ;
            else break ;
        }
        cou_mots=cou_mots->suivant ;
    }
}

```

```

    }
    if (!dn_mots) return(pre_mots) ;

    /* recherche du dernier Dnum */

    cou_mots=dn_mots->suivant ;
    while (cou_mots)
    {
        test=0 ;
        if ((cou_mots->suivant)&&
            (!alpha(cou_mots->mot, "et"))&&
            ((!alpha((cou_mots->suivant)->mot, "un"))||
            (!alpha((cou_mots->suivant)->mot, "une"))||
            (!alpha((cou_mots->suivant)->mot, "unième"))||
            (!alpha((cou_mots->suivant)->mot, "unièmes"))))) test=2 ;
        else
        {
            cou_dico=cou_mots->motdico ;
            while (cou_dico)
            {
                if ((!comp_ch(cou_dico->temps, "DN"))&&
                    (((cou_dico->mot[0]!='d')&&(cou_dico->mot[1]!='n'))&&
                    ((cou_dico->mot[0]!='r')&&(cou_dico->mot[1]!='m'))))
                {
                    um_mots=cou_mots ;
                    test=1 ;
                    break ;
                }
                if (!comp_ch(cou_dico->temps, "EM"))
                {
                    um_mots=cou_mots ;
                    testieme=1 ;
                    break ;
                }
                cou_dico=cou_dico->suivant ;
            }
        }
        if (!test) break ;
        cou_mots=cou_mots->suivant ;
    }
    if (!um_mots) return(pre_mots) ;

    /* dimension du tableau */

    cou_mots=dn_mots ;
    while ((cou_mots=cou_mots->suivant)!=um_mots) dim++ ;
    if (dim>MAXDNUM)
    {
        ecrire("\nRevoir la définition de MAXDNUM dans le fichier BIBLI.H") ;
        termine(1) ;
    }

    /* remplissage du tableau */
    /* "et" prend la valeur -1 */

    cou_mots=dn_mots ;
    for (i=0 ; i<dim ; i++)
    {

```

```

        if ((!alpha(cou_mots->mot, "cent")) ||
(!alpha(cou_mots->mot, "cents")) ||
(!alpha(cou_mots->mot, "centième")))
    || (!alpha(cou_mots->mot, "centièmes"))) tableau[i] = 100 ;
        if ((!alpha(cou_mots->mot, "cinq")) ||
(!alpha(cou_mots->mot, "cinquième")) ||
(!alpha(cou_mots->mot, "cinquièmes"))) tableau[i] = 5 ;
        if ((!alpha(cou_mots->mot, "cinquante")) ||
(!alpha(cou_mots->mot, "cinquantième")) ||
(!alpha(cou_mots->mot, "cinquantièmes"))) tableau[i] = 50;
        if ((!alpha(cou_mots->mot, "deux")) ||
(!alpha(cou_mots->mot, "deuxième")) ||
(!alpha(cou_mots->mot, "deuxièmes"))) tableau[i] = 2 ;
        if ((!alpha(cou_mots->mot, "dix")) ||
(!alpha(cou_mots->mot, "dixième")) ||
(!alpha(cou_mots->mot, "dixièmes"))) tableau[i] = 10 ;
        if ((!alpha(cou_mots->mot, "douze")) ||
(!alpha(cou_mots->mot, "douzième")) ||
(!alpha(cou_mots->mot, "douzièmes"))) tableau[i] = 12 ;
        if (!alpha(cou_mots->mot, "et")) tableau[i] = -1 ;
        if ((!alpha(cou_mots->mot, "huit")) ||
(!alpha(cou_mots->mot, "huitième")) ||
(!alpha(cou_mots->mot, "huitièmes"))) tableau[i] = 8 ;
        if ((!alpha(cou_mots->mot, "huitante")) ||
(!alpha(cou_mots->mot, "huitantième")) ||
(!alpha(cou_mots->mot, "huitantièmes"))) tableau[i] = 80 ;
        if ((!alpha(cou_mots->mot, "mil")) ||
(!alpha(cou_mots->mot, "mille")) ||
(!alpha(cou_mots->mot, "millième")) ||
(!alpha(cou_mots->mot, "millièmes"))) tableau[i] = 1000 ;
        if ((!alpha(cou_mots->mot, "neuf")) ||
(!alpha(cou_mots->mot, "neuvième")) ||
(!alpha(cou_mots->mot, "neuvièmes"))) tableau[i] = 9 ;
        if ((!alpha(cou_mots->mot, "nonante")) ||
(!alpha(cou_mots->mot, "nonantième")) ||
(!alpha(cou_mots->mot, "nonantièmes"))) tableau[i] = 90 ;
        if ((!alpha(cou_mots->mot, "octante")) ||
(!alpha(cou_mots->mot, "octantième")) ||
(!alpha(cou_mots->mot, "octantièmes"))) tableau[i] = 80 ;
        if ((!alpha(cou_mots->mot, "onze")) ||
(!alpha(cou_mots->mot, "onzième")) ||
(!alpha(cou_mots->mot, "onzièmes"))) tableau[i] = 11 ;
        if ((!alpha(cou_mots->mot, "quarante")) ||
(!alpha(cou_mots->mot, "quarantième")) ||
(!alpha(cou_mots->mot, "quarantièmes"))) tableau[i] = 40 ;
        if ((!alpha(cou_mots->mot, "quatorze")) ||
(!alpha(cou_mots->mot, "quatorzième")) ||
(!alpha(cou_mots->mot, "quatorzièmes"))) tableau[i] = 14 ;
        if ((!alpha(cou_mots->mot, "quatre")) ||
(!alpha(cou_mots->mot, "quatrième")) ||
(!alpha(cou_mots->mot, "quatrièmes"))) tableau[i] = 4 ;
        if ((!alpha(cou_mots->mot, "quinze")) ||
(!alpha(cou_mots->mot, "quinzième")) ||
(!alpha(cou_mots->mot, "quinzièmes"))) tableau[i] = 15 ;
        if ((!alpha(cou_mots->mot, "seize")) ||
(!alpha(cou_mots->mot, "seizième")) ||
(!alpha(cou_mots->mot, "seizièmes"))) tableau[i] = 16 ;

```

```

        if ((!alpha(cou_mots->mot, "sept")) ||
(!alpha(cou_mots->mot, "septième")) ||
(!alpha(cou_mots->mot, "septièmes"))) tableau[i] = 7 ;
        if ((!alpha(cou_mots->mot, "septante")) ||
(!alpha(cou_mots->mot, "septantième")) ||
(!alpha(cou_mots->mot, "septantièmes"))) tableau[i] = 70 ;
        if ((!alpha(cou_mots->mot, "six")) ||
(!alpha(cou_mots->mot, "sixième")) ||
(!alpha(cou_mots->mot, "sixièmes"))) tableau[i] = 6 ;
        if ((!alpha(cou_mots->mot, "soixante")) ||
(!alpha(cou_mots->mot, "soixantième")) ||
(!alpha(cou_mots->mot, "soixantièmes"))) tableau[i] = 60 ;
        if ((!alpha(cou_mots->mot, "treize")) ||
(!alpha(cou_mots->mot, "treizième")) ||
(!alpha(cou_mots->mot, "treizièmes"))) tableau[i] = 13 ;
        if ((!alpha(cou_mots->mot, "trente")) ||
(!alpha(cou_mots->mot, "trétième")) ||
(!alpha(cou_mots->mot, "trentièmes"))) tableau[i] = 30 ;
        if ((!alpha(cou_mots->mot, "trois")) ||
(!alpha(cou_mots->mot, "troisième")) ||
(!alpha(cou_mots->mot, "troisièmes"))) tableau[i] = 3 ;
        if ((!alpha(cou_mots->mot, "un")) ||
(!alpha(cou_mots->mot, "unième")) ||
(!alpha(cou_mots->mot, "unièmes"))) tableau[i] = 1 ;
        if ((!alpha(cou_mots->mot, "vingt")) ||
(!alpha(cou_mots->mot, "vingtième")) ||
(!alpha(cou_mots->mot, "vingtièmes"))) tableau[i] = 20 ;
        if (!alpha(cou_mots->mot, "zéro")) tableau[i] = 0 ;
        cou_mots = cou_mots->suivant ;
    }

    /* contraction des "0" et des "et" */

    for (i=0 ; i<(dim-1)&&(dim>1) ; i++) if (tableau[i]<=0)
    {
        for (j=i-- ; j<dim ; j++) tableau[j]=tableau[j+1] ;
        dim-- ;
    }

    /* contraction de "17", "18", "19" */
    /* (à cause de 1700, 1800, 1900) */

    for (i=0 ; i<(dim-1)&&(dim>1) ; i++)
    if ((tableau[i]==10)&&
((tableau[i+1]==7) || (tableau[i+1]==8) || (tableau[i+1]==9)))
    {
        tableau[i+1] += tableau[i] ;
        for (j=i-- ; j<dim ; j++) tableau[j]=tableau[j+1] ;
        dim-- ;
    }

    /* contraction si x<y => x*y */

    for (i=0 ; i<(dim-1)&&(dim>1) ; i++)
    if (tableau[i]<tableau[i+1])
    {
        tableau[i+1]*=tableau[i] ;
        for (j=i-- ; j<dim ; j++)
            tableau[j]=tableau[j+1] ;
    }

```

```

        dim-- ;
    }
        /* contraction si x>y => x+y */
for (i=0 ; (i<(dim-1))&&(dim>1) ; i++)
    if (tableau[i]>tableau[i+1])
    {
        tableau[i+1] += tableau[i] ;
        for (j=i-- ; j<dim ; j++)
            tableau[j]=tableau[j+1] ;
        dim-- ;
    }
        /* libération des champs DICO */
        /* suppression des décalage */

cou_mots=dn_mots ;
while ((cou_mots)&&(cou_mots->precedent!=um_mots))
{
    cou_dico=cou_mots->motdico ;
    do libere(cou_dico) ;
    while ((cou_dico=cou_dico->suivant)) ;
    cou_mots->motdico=RIEN ;
    cou_mots->decale=RIEN ;
    cou_mots=cou_mots->suivant ;
}
dn_mots->decale=um_mots->suivant ;

        /* sortie des "ième" */

if (testieme)
{
    dim=l_ch(um_mots->mot) ;
    if (um_mots->mot[dim-1]='s')
        dn_mots->motdico=cp_dico("èmes", "A31:Amp:Afp", "MO") ;
    else dn_mots->motdico=cp_dico("ème", "A31:Ams:Afs", "MO") ;
}
else
{
    dn_mots->motdico=cp_dico("dnum", "A80P:Amp:Afp", "DN") ;
    cou_dico=dn_mots->motdico ;
    if ((tableau[0]>99)&&(tableau[0]<10000))
    {
        cou_dico->suivant=cp_dico("dnumannee", "A80P:Amp:Afp",
"DA") ;
        cou_dico=cou_dico->suivant ;
    }
    if (tableau[0]<100)
    {
        cou_dico->suivant=cp_dico("dnuman", "A80P:Amp:Afp", "DB") ;
        cou_dico=cou_dico->suivant ;
    }
    if (tableau[0]<25)
    {
        cou_dico->suivant=cp_dico("dnumheure", "A80P:Amp:Afp",
"DH") ;
        cou_dico=cou_dico->suivant ;
    }
}

```

```

if (tableau[0]<32)
{
    cou_dico->suivant=cp_dico("dnumjour", "A80P:Amp:Afp", "DJ") ;
    cou_dico=cou_dico->suivant ;
}
if (tableau[0]<60)
{
    cou_dico->suivant=cp_dico("dnumminute", "A80P:Amp:Afp",
"DK") ;
    cou_dico=cou_dico->suivant ;
}
if (tableau[0]<13)
{
    cou_dico->suivant=cp_dico("dnummois", "A80P:Amp:Afp", "DM")
;
    cou_dico=cou_dico->suivant ;
}
if (!tableau[0]%10)
{
    cou_dico->suivant=cp_dico("dnumdecennie", "A80P:Amp:Afp",
"DO") ;
    cou_dico=cou_dico->suivant ;
}
}

/* itération éventuelle */

if (um_mots->suivant)
    return(dnum(pre_mots, um_mots->suivant)) ;
return(pre_mots) ;
}

```

9.5. FICHIER *MODULE2.C*

```

#include "definir.h"

extern sort_imp ;           /* sortie sur imprimante */
extern MOTS *module2 ;     /* état du module 2 */

/* sélectionne le meilleur CHOIX possible */
/* retourne la liste à partir du premier MOTS reconnu */
/* ou RIEN si aucun MOTS n'est reconnu */
/* imprime s'il n'y a pas d'état final */

MOTS *compauto(pre_mots)

MOTS *pre_mots ;           /* premier MOTS à étudier */
{
    int meilleur,           /* nombre de MOTS du meilleur CHOIX */
        courant,           /* et du CHOIX courant */
        i ;                /* indentation */
    char mot[MAXMOT+1] ;
    MOTS *deb_mots, *cou_mots,
        *der_mots ;        /* MOTS à renvoyer */
    static MOTS *rep_mots=RIEN ; /* reprise immédiate */
    DICO *cou_dico ;
    CHOIX *pre_choi, *cou_choi,
        *mei_choi ;        /* meilleur choix (le plus long) */
    extern CHOIX *clf_choi(), *droite() ;
    extern testauto(), comptnoe(), alphacct() ;
    extern void ecran(), imprime() ;
    extern MOTS *lib_dico(), *passage() ;

    if (rep_mots) deb_mots=rep_mots ;
    else
    {
        if (module2) /* le premier MOTS est non reconnu */
        {
            cou_mots=module2 ;
            module2=RIEN ;
            while ((cou_mots)&&(!cou_mots->motdico))
            {
                ecran(cou_mots->mot, 'G') ;
                ecran(cou_mots->car, 'G') ;
                if (sort_imp)
                {
                    imprime(cou_mots->mot, 'C') ;
                    imprime(cou_mots->car, 'C') ;
                }
                libere(cou_mots) ;
                cou_mots=cou_mots->suivant ;
            }
            if (!cou_mots) return(RIEN) ;
            pre_mots=cou_mots ;
            pre_mots->precedent=RIEN ;
            deb_mots=pre_mots ;
        }
        else deb_mots=pre_mots ;
        if (!deb_mots) return(RIEN) ;
    }
}

```

```

if (!testauto(deb_mots))
{
    cou_dico = deb_mots->motdico ;
    do libere(cou_dico) ;
    while ((cou_dico = cou_dico->suivant)) ;
    deb_mots->motdico = RIEN ;
    module2 = deb_mots ;
    return(compauto(deb_mots->decale)) ;
}

pre_choi = clf_choi(deb_mots) ;
if (!pre_choi)
{
    cou_dico = deb_mots->motdico ;
    do libere(cou_dico) ;
    while ((cou_dico = cou_dico->suivant)) ;
    deb_mots->motdico = RIEN ;
    module2 = deb_mots ;
    if (rep_mots)
    {
        rep_mots = RIEN ;
        cou_mots = deb_mots->precedent ;
        cou_mots->suivant = RIEN ;
        while (!cou_mots->motdico)
            cou_mots = cou_mots->precedent ;
        cou_mots->decale = RIEN ;
        cou_mots = deb_mots->precedent ;
        cou_dico = cou_mots->motdico ;
        if (!comp_ch(cou_dico->temps, "DE"))
        {
            ecran(cou_mots->mot, 'G') ;
            ecran(cou_mots->car, 'G') ;
            if (sort_imp)
            {
                imprime(cou_mots->mot, 'C') ;
                imprime(cou_mots->car, 'C') ;
            }
            libere(cou_dico) ;
            libere(cou_mots) ;
        }
        module2->precedent = RIEN ;
        return(pre_mots) ;
    }
    return(compauto(deb_mots->decale)) ;
}

cou_choi = droite(pre_choi) ;
mei_choi = cou_choi ;
meilleur = comptnoe(cou_choi, pre_choi) ;
while (!meilleur) /* meilleur=0 <=> état non final */
{
    if (cou_choi->gauche)
    {
        cou_choi = cou_choi->gauche ;
        if (cou_choi->fils)
            cou_choi = droite(cou_choi->fils) ;
    }
}

```

```

else
{
    if (cou_choi->pere)
        cou_choi=cou_choi->pere ;
    else break ;
}
mei_choi=cou_choi ;
meilleur=comptnoe(cou_choi, pre_choi) ;
}

while (cou_choi!=pre_choi)
{
    while (cou_choi->pere)
        cou_choi=cou_choi->pere ;
    if (cou_choi->gauche)
    {
        cou_choi=cou_choi->gauche ;
        if (cou_choi->fils)
            cou_choi=droite(cou_choi->fils) ;
    }
    else break ;
    courant=comptnoe(cou_choi, pre_choi) ;
    while (!courant)
    {
        if (cou_choi->gauche)
        {
            cou_choi=cou_choi->gauche ;
            if (cou_choi->fils)
                cou_choi=droite(cou_choi->fils) ;
        }
        else
        {
            if (cou_choi->pere)
                cou_choi=cou_choi->pere ;
            else break ;
        }
        courant=comptnoe(cou_choi, pre_choi) ;
    }
    if (meilleur<courant)
    {
        mei_choi=cou_choi ;
        meilleur=courant ;
    }
}
if (meilleur) /* on libère les DICO inutiles */
{
    cou_mots=deb_mots ;
    for (i=1 ; i<meilleur ; i++)
        cou_mots=cou_mots->decale ;
    der_mots=cou_mots->decale ;
    cou_choi=mei_choi ;
    for (i=meilleur ; i>0 ; i--)
    {
        lib_dico(cou_mots, (cou_choi->reconnu)->temps) ;
        do_cou_mots=cou_mots->precedent ;
        while (!cou_mots->motdico) ;
        while (cou_choi->gauche)
            cou_choi=cou_choi->gauche ;
    }
}

```

```

        cou_choi=cou_choi->pere ;
    }
}

/* libération des CHOIX */

cou_choi=droite(pre_choi) ;
while (cou_choi!=pre_choi)
{
    while (cou_choi->pere)
    {
        libere(cou_choi) ;
        cou_choi=cou_choi->pere ;
    }
    if (cou_choi->gauche)
    {
        libere(cou_choi) ;
        cou_choi=cou_choi->gauche ;
        if (cou_choi->fils)
            cou_choi=droite(cou_choi->fils) ;
    }
}
libere(pre_choi) ;

/* s'il n'y a pas d'état final */
/* on renvoie le premier MOTS */

if (!meilleur)
{
    cou_dico=deb_mots->motdico ;
    do libere(cou_dico) ;
    while ((cou_dico=cou_dico->suivant)) ;
    deb_mots->motdico=RIEN ;
    module2=deb_mots ;
    if (rep_mots)
    {
        rep_mots=RIEN ;
        cou_mots=deb_mots->precedent ;
        cou_mots->suivant=RIEN ;
        while (!cou_mots->motdico)
            cou_mots=cou_mots->precedent ;
        cou_mots->decale=RIEN ;
        module2->precedent=RIEN ;
        return(pre_mots) ;
    }
    return(compauto(deb_mots->decale)) ;
}

/* adverbess reliés par la préposition "de" */

if (der_mots->decale)
{
    cou_dico=der_mots->motdico ;
    if (((!alpha(cou_dico->mot, "d"))&&(!alpha(der_mots->car,
"")))) || (!alpha(cou_dico->mot, "de")))
    {
        der_mots=passage(der_mots, "DE") ;
        der_mots=der_mots->decale ;
    }
}

```

```

if (!alpha(cou_dico->mot, "des"))
{
    der_mots=passage(der_mots, "DELS");
    der_mots=der_mots->decale;
}
if (!alpha(cou_dico->mot, "du"))
{
    der_mots=passage(der_mots, "DELE");
    der_mots=der_mots->decale;
}
/* adverbess reliés par la préposition "à" */

if (!alpha(cou_dico->mot, "à"))
{
    der_mots=passage(der_mots, "PR");
    der_mots=der_mots->decale;
}
if (!alpha(cou_dico->mot, "au"))
{
    der_mots=passage(der_mots, "PRLE");
    der_mots=der_mots->decale;
}
if (!alpha(cou_dico->mot, "aux"))
{
    der_mots=passage(der_mots, "PRLS");
    der_mots=der_mots->decale;
}
/* adverbess reliés par les conjonctions "et", "ni" et "ou" */

if ((!alpha(cou_dico->mot, "et")) || (!alpha(cou_dico->mot,
"ni"))) || (!alpha(cou_dico->mot, "ou")))
{
    der_mots=passage(der_mots, "CJ");
    der_mots=der_mots->decale;
}
}

/* itération avec marquage ('&') */

if (der_mots)
{
    cou_dico=der_mots->motdico;
    cop_ch(mot, "&");
    cat_ch(mot, cou_dico->mot);
    cop_ch(cou_dico->mot, mot);
    rep_mots=der_mots;
    return(compauto(pre_mots));
}
rep_mots=RIEN;
module2=RIEN;
return(pre_mots);
}

```

9.6. FICHER *MODULE3.C*

```

#include "definir.h"

extern sort_imp, /* sortie sur imprimante */
sort_res; /* sortie sur le fichier des résidus */
extern FICHER *fic_res; /* fichier des résidus */
extern MOTS *module3; /* état du module 3 */

/* gère la consultation des tables de transitions */
/* imprime les mots à éliminer */
/* renvoie la liste des MOTS reconnus */

MOTS *tabtrans(pre_mots)

MOTS *pre_mots; /* premier MOTS à étudier */
{
    static char lst_tps[MAXLIST*2+1]; /* chaîne des champs temps */
    int longueur, /* d'une chaîne puis du tableau */
    i,j; /* indentation */
    static MOTS *lst_mots[MAXLIST]; /* tableau de */
    /* pointeurs vers les MOTS de la liste */
    MOTS *cou_mots;
    DICO *cou_dico;
    extern char *contr_do(), *contr_nt(), *contr_ca(),
    *contr_mo(), *contr_pd(), *contr_dd(),
    *contr_dt(), *contr_gp(), *contr_le(),
    *contr_de(), *contr_di(), *contr_er(),
    *contrali();
    extern void ecran(), imprime();
    if (!module3)
    {
        if (!pre_mots) return(RIEN);

        /* remplissage de la chaîne et du tableau */

        if (sort_res) fprintf(fic_res, "\\");
        cou_mots = pre_mots;
        for (i=0; (cou_mots)&&(i<MAXLIST); i++)
        {
            lst_mots[i] = cou_mots;
            cou_dico = cou_mots->motdico;
            longueur = 1 + ch(cou_dico->temps);
            lst_tps[2*i] = (cou_dico->temps)[0];
            lst_tps[2*i+1] = (cou_dico->temps)[1];
            if (longueur == 4)
            {
                lst_mots[i+i] = cou_mots;
                lst_tps[2*i] = (cou_dico->temps)[2];
                lst_tps[2*i+1] = (cou_dico->temps)[3];
            }
            cou_mots = cou_mots->decale;
        }
        if ((i == MAXLIST)&&(cou_mots))
        {
            ecrire("\nRevoir la définition de MAXLIST dans le fichier
DEFINIR.H");
        }
    }
}

```

```

        termine(2) ;
    }
    lst_tps[2*i]='\0' ;

        /* contraction des ères */
    cop_ch(lst_tps, contr_er(lst_tps, lst_mots, pre_mots)) ;

        /* contraction des modificateurs-années */
    cop_ch(lst_tps, contr_do(lst_tps, lst_mots, pre_mots)) ;

        /* contraction des modificateurs-Ntps */
    cop_ch(lst_tps, contr_nt(lst_tps, lst_mots, pre_mots)) ;

        /* contraction des Dnum et demi */
    cop_ch(lst_tps, contr_di(lst_tps, lst_mots)) ;

        /* contraction des dates */
    cop_ch(lst_tps, contr_ca(lst_tps, lst_mots)) ;

        /* étude des modificateurs */
    cop_ch(lst_tps, contr_mo(lst_tps, lst_mots, pre_mots)) ;

        /* préposition et déterminant */
    cop_ch(lst_tps, contr_pd(lst_tps, lst_mots, pre_mots)) ;

        /* combinaison de déterminants */
    cop_ch(lst_tps, contr_dd(lst_tps, lst_mots, pre_mots)) ;

        /* contraction des déterminants */
    cop_ch(lst_tps, contr_dt(lst_tps, lst_mots, pre_mots)) ;

        /* contraction des groupes prépositionnels */
    cop_ch(lst_tps, contr_gp(lst_tps, lst_mots, pre_mots)) ;

        /* contraction de LE et CE */
    cop_ch(lst_tps, contr_le(lst_tps, lst_mots)) ;

        /* contraction des différents adverbes liés par "de" */
    cop_ch(lst_tps, contr_de(lst_tps, lst_mots)) ;
}
else pre_mots=module3 ;

```

```

/* écriture des premiers MOTS non reconnus */

while (lst_tps[0] == '$')
{
    longueur = 1 ch(lst_tps)/2 ;
    cou_mots = lst_mots[0] ;
    while (cou_mots->precedent)
        cou_mots = cou_mots->precedent ;
    pre_mots = cou_mots ;
    cou_dico = cou_mots->motdico ;
    while (((longueur != 1) && (cou_mots != lst_mots[1])))
    || ((longueur == 1) && (cou_mots)))
    {
        ecran(cou_mots->mot, 'V') ;
        ecran(cou_mots->car, 'V') ;
        if (sort_imp)
        {
            imprime(cou_mots->mot, 'N') ;
            imprime(cou_mots->car, 'N') ;
        }
        if (sort_res) fprintf(fic_res, "%s%s", cou_mots->mot, cou_mots-
> car) ;
        libere(cou_dico) ;
        libere(cou_mots) ;
        cou_mots = cou_mots->suivant ;
        if (cou_mots)
        {
            cou_dico = cou_mots->motdico ;
            if ((cou_dico->mot[0] == '&') ||
                (cou_dico->mot[1] == '&')) break ;
        }
        if (cou_mots) cou_mots->precedent = RIEN ;
        cop_ch7(lst_tps, contrali(lst_tps, lst_mots, 0)) ;
    }
    longueur = 1 ch(lst_tps)/2 ;
    if (!longueur)
    {
        module3 = RIEN ;
        return(RIEN) ;
    }
}

/* mots reconnus */

cou_mots = lst_mots[0] ;
while (cou_mots->precedent)
    cou_mots = cou_mots->precedent ;
pre_mots = cou_mots ;

for (i = 1 ; i < longueur ; i++)
    if (lst_tps[2*i] == '$') break ;
if (i == longueur)
{
    module3 = RIEN ;
    return(pre_mots) ;
}

cou_mots = lst_mots[i] ;
cou_dico = cou_mots->motdico ;

```

```

    while (((cou_dico->mot[0]!='&') || (cou_dico->mot[1]=='&')) && (cou_mots-
>precedent != lst_mots[i-1]))
    {
        cou_mots=cou_mots->precedent ;
        cou_dico=cou_mots->motdico ;
    }
    module3=cou_mots ;
    cou_mots=cou_mots->precedent ;
    cou_mots->suivant=RIEN ;
    while (!cou_mots->motdico)
        cou_mots=cou_mots->precedent ;
    cou_mots->decale=RIEN ;
    module3->precedent=RIEN ;

    for (j=0 ; j<i ; j++)
        cop_ch(lst_tps, contrali(lst_tps, lst_mots, 0)) ;

    return(pre_mots) ;
}

```

9.7. FICHIER *ASCII.DOS.C*

```
# include "definir.h"

        /***** code ASCII *****/

        /* convertit c en majuscule */
        /* utilise le code ASCII (formule: code-256) */

char majus(c)

int c ;
{
    if ((c == -125) || (c == -124) || (c == -123) || (c == -114))
        return('A') ;
    if ((c == -126) || (c == -120) || (c == -119) || (c == -118) || (c == -112))
        return('E') ;
    if ((c == -117) || (c == -116)) return('I') ;
    if ((c == -109) || (c == -108) || (c == -103)) return('O') ;
    if ((c == -127) || (c == -106) || (c == -105) || (c == -102))
        return('U') ;
    if ((c == -104)) return('Y') ;
    if ((c == -128) || (c == -121)) return('C') ;

    if ((c >= 97) && (c <= 122)) return((c - 32)) ;

    return(c) ;
}

        /* détermine si un caractère est une lettre ou un chiffre */
        /* retournent 1 -> chiffres */
        /* 2 -> majuscules */
        /* 3 -> minuscules */
        /* 4 -> minuscules accentuées */

est_ltnb(c)

int c ;
{
    if ((c >= 48) && (c <= 57)) return(1) ;
    if ((c >= 65) && (c <= 90)) return(2) ;
    if ((c >= 97) && (c <= 122)) return(3) ;
    if ((c >= -128) && (c <= -102)) return(4) ;
}

        /***** gestion d'écran *****/

        /* écrit une chaîne à l'écran */

void ecrire(chaine)

char chaine[] ;
{
    for (; *chaine ; chaine++) ec_car_e(*chaine) ;
}
```

/* écrire à l'écran à une position donnée */

void posit_e(chaine, ligne, colonne)

```
char chaine[] ;
int ligne, colonne ;
{
    char tampon[255], ch_li[3], ch_co[3] ;
    extern void ecrire() ;

    if ((ligne < 0) || (ligne > 24) || (colonne < 0) || (colonne > 79))
        ecrire(chaine) ;
    else
    {
        sprintf(ch_li, "%d", ligne) ;
        sprintf(ch_co, "%d", colonne) ;
        cop_ch(tampon, "\x1B[");
        cat_ch(tampon, ch_li) ;
        cat_ch(tampon, ";");
        cat_ch(tampon, ch_co) ;
        cat_ch(tampon, "H") ;
        cat_ch(tampon, chaine) ;
        ecrire(tampon) ;
    }
}
```

/* écrit en gras (surbrillance), clignotant */
 /* souligné (adaptateur monochrome seulement) */
 /* ou vidéo inverse: attribut=G, C, S, I */
 /* combinaison vidéo inverse et gras: attribut=V */

void ecran(chaine, attribut)

```
char chaine[] ;
int attribut ;
{
    char resultat[MAXCHAIN + 1] ;
    extern void ecrire() ;

    cop_ch(resultat, "\x1B[");
    switch (attribut)
    {
        case 'G' : cat_ch(resultat, "1m") ;
                    break ;

        case 'S' : cat_ch(resultat, "4m") ;
                    break ;

        case 'C' : cat_ch(resultat, "5m") ;
                    break ;

        case 'I' : cat_ch(resultat, "7m") ;
                    break ;

        case 'V' : cat_ch(resultat, "7m\x1B[1m") ;
                    break ;

        default : ecrire(chaine) ;
    }
}
```

```

    }
    cat_ch(resultat, chaine);
    cat_ch(resultat, "\x1B[0m");
    ecrire(resultat);
}

/* effacement de l'écran */

void efface_e()
{
    extern void ecrire();

    ecrire("\x1B[2J");
}

/****** gestion d'imprimante *****/

/* écriture sur l'imprimante en caractères normaux (N) */
/* élargis (E), compressés (C), gras (G), double frappe (D) */
/* exposants (X), indices (I) ou soulignés (S) */

void imprime(chaine, attribut)
char chaine[];
int attribut;
{
    char resultat[MAXCHAIN + 1];
    int impriman;

    impriman = ouvre_ec("PRN");
    switch (attribut)
    {
        case 'E' : cop_ch(resultat, "\x0E");
                   cat_ch(resultat, chaine);
                   cat_ch(resultat, "\x14");
                   break;

        case 'C' : cop_ch(resultat, "\x0F");
                   cat_ch(resultat, chaine);
                   cat_ch(resultat, "\x12");
                   break;

        case 'G' : cop_ch(resultat, "\x1BE");
                   cat_ch(resultat, chaine);
                   cat_ch(resultat, "\x1BF");
                   break;

        case 'D' : cop_ch(resultat, "\x1BG");
                   cat_ch(resultat, chaine);
                   cat_ch(resultat, "\x1BH");
                   break;

        case 'X' : cop_ch(resultat, "\x1BS0");
                   cat_ch(resultat, chaine);
                   cat_ch(resultat, "\x1BT");
                   break;
    }
}

```

```

        case 'I' : cop_ch(resultat, "\x1BS1") ;
                    cat_ch(resultat, chaine) ;
                    cat_ch(resultat, "\x1BT") ;
                    break ;

        case 'S' : cop_ch(resultat, "\x1B-1") ;
                    cat_ch(resultat, chaine) ;
                    cat_ch(resultat, "\x1B-0") ;
                    break ;

        default : cop_ch(resultat, chaine) ;
    }
    ecrit(impriman, resultat, l_ch(resultat)) ;
    ferme(impriman) ;
    return ;
}

/* fin de l'impression (vidage du buffer) */

void fin_impr()
{
    int impriman ;

    impriman = ouvre_ec("PRN") ;
    ecrit(impriman, "\x0D\x0A", 2) ;
    ferme(impriman) ;
    return ;
}

/* initialisation de l'imprimante */

void ini_impr()
{
    int impriman ;

    impriman = ouvre_ec("PRN") ;
    ecrit(impriman, "\x1B\x40", 2) ;
    ferme(impriman) ;
    return ;
}

```

9.8. FICHIER *OUTILS.C*

```

#include "definir.h"

extern sort_imp, /* sortie sur imprimante */
sort_res; /* sortie sur le fichier des résidus */
extern FICHIER *fic_indx, /* fichier index */
*fic_mots, /* fichier dictionnaire */
*fic_tran; /* fichier automate */

/* ***** gestion de fichiers ***** */

/* charge un fichier */

FICHIER *ouvertur(nomfich, mode)

char nomfich[];
int mode; /* 1-> lecture, 0-> ajouts en fin -1-> création */
{
    FICHIER *fiche;

    if (!mode) fiche = ouv_a_f(nomfich);
    else
    {
        if (mode > 0) fiche = ouv_l_f(nomfich);
        else fiche = ouv_ec_f(nomfich);
    }
    if (!fiche)
    {
        printf("\nIl y a erreur dans l'ouverture du fichier %s!", nomfich);
        termine(3);
    }
    return(fiche);
}

/* ferme un fichier */

void fermetur(fiche)

FICHIER *fiche;
{
    extern void ecrire();

    if ((ferme f(fiche)))
        ecrire("\nAttention! erreur dans la fermeture des fichiers!");
}

/* ***** alphabet ***** */

/* comparaison alphabétique des deux */

alpha(ch1, ch2)

/* chaînes mises en majuscules : 0 si =, >0 si x>y */
char ch1[], ch2[];
{
    char maj1[MAXCHAIN + 1], maj2[MAXCHAIN + 1];
    int i, longueur;

```

```

extern char majus() ;

longueur=l_ch(ch1) ;
for (i=0 ; i<longueur ; i++) maj1[i]=majus(ch1[i]) ;
maj1[i]='\0' ;
longueur=l_ch(ch2) ;
for (i=0 ; i<longueur ; i++) maj2[i]=majus(ch2[i]) ;
maj2[i]='\0' ;
return(comp_ch(maj1, maj2)) ;
}

```

```

/* test si c est une minuscule */
/* retourne 2, 0 sinon */

```

est_minu(c)

```

char c;
{
    extern char majus() ;

    if (!est_lt(majus(c))) return(0) ;
    if (!est_maj(c)) return(2) ;
    else return(0) ;
}

```

```

/* suite de alpha : gestion des accents */
/* ne tient pas compte des majuscules */

```

alphacct(ch1, ch2)

```

char ch1[], ch2[] ;
{
    int resultat ; /* 0 si =, >0 si x>y */
    char ch12[MAXCHAIN+1], ch22[MAXCHAIN+1] ;
    extern alpha() ;
    extern char *maj_min() ;

    resultat=alpha(ch1,ch2) ;
    if (!resultat)
    {
        cop_ch(ch12, maj_min(ch1)) ;
        cop_ch(ch22, maj_min(ch2)) ;
        resultat=comp_ch(ch12, ch22) ;
    }
    return(resultat) ;
}

```

```

/* transforme une chaîne de */
/* caractères en minuscules */

```

char *maj_min(chaine)

```

char chaine[] ;
{
    char resultat[MAXCHAIN+1] ;
    int i, l ;

    l=l_ch(chaine) ;

```

```

    for (i=0 ; i<1 ; i++)
        resultat[i]=minuscule(chaine[i]) ;
    resultat[i]='\0' ;
    return(resultat) ;
}

/* renvoie une partie d'une chaîne */

char *part_ch(totale, i, j)

char totale[] ;          /* chaîne de départ */
int i,                   /* premier caractère à prendre */
    j ;                  /* dernier caractère à prendre */
{
    int k ;              /* indentation */
    char ch[MAXCHAIN] ;

    if ((!totale[0]) || (j<i) || (l_ch(totale)<=j)) return(""); ;

    for (k=i ; k<=j ; k++) ch[k-i]=totale[k] ;
    ch[k-i]='\0' ;
    return(ch) ;
}

/* test sur les caractères d'une chaîne */
/* retourne: A pour chiffres->minuscules */
/*          B pour chiffres */
/*          C pour minuscules */
/*          D pour majuscules->minuscules */
/*          E pour majuscules */
/*          X sinon */
/* suivi d'un caractère indiquant le nombre */
/* de chiffres ou de majuscules pour A et D */

char *test_car(chaine)

char chaine[] ;
{
    int longueur,        /* de la chaîne */
        i,               /* indentation */
        nombre ;         /* de caractères de première forme */
    char resultat[5],     /* une lettre, trois chiffres */
        test,            /* une lettre */
        nombrech[4] ;    /* chaîne=nombre */
    extern char majus() ;
    extern est_minu(), est_ltnb() ;

    nombre=0 ;
    longueur=l_ch(chaine) ;

    /* premier caractère */

    if (!est_ltnb(chaine[0])) return("X") ;
    if (est_maj(chaine[0]))
    {
        test='E' ;
        nombre=1 ;
    }
}

```

```

if (est_minu(chaine[0])) test='C' ;
if (est_nb(chaine[0]))
{
    test='B' ;
    nombre = 1 ;
}

/* caractères suivants */

for (i=1 ; i<longueur ; i++)
{
    if (!est_ltnb(chaine[i])) return("X") ;
    if (test=='E')
    {
        if (!est_lt(majus(chaine[i]))) return("X") ;
        if (est_maj(chaine[i])) nombre += 1 ;
        if (est_minu(chaine[i])) test='D' ;
    }
    if ((test=='D')&&(!est_minu(chaine[i]))) return("X");
    if ((test=='C')&&(!est_minu(chaine[i]))) return("X");
    if (test=='B')
    {
        if (est_nb(chaine[i])) nombre += 1 ;
        if (est_minu(chaine[i])) test='A' ;
        if (est_maj(chaine[i])) return("X") ;
    }
    if ((test=='A')&&(!est_minu(chaine[i]))) return("X");
}

/* résultat */

if ((test=='B')|| (test=='C') || (test=='E'))
{
    resultat[0]=test ;
    resultat[1]='\0' ;
    return(resultat) ;
}
resultat[0]=test ;
resultat[1]=nombre ;
resultat[2]='\0' ;
return(resultat) ;
}

/* transforme une chaîne en minuscule */
/* suivant le résultat de test_car */
/* A à C -> pas de transformation */
/* D -> met une majuscule en première place */
/* E -> transforme tout en majuscule */

char *min_maj(ch, format)

char ch[MAXCHAIN+1], /* chaîne à transformer */
    format ;          /* C, D ou E */
{
    int longueur, /* de la chaîne */
        i ;       /* indentation */
    extern char majus() ;

```

```

switch (format)
{
    case 'A' : return(ch) ;

    case 'B' : return(ch) ;

    case 'C' : return(ch) ;

    case 'D' : ch[0]=majus(ch[0]) ;
                return(ch) ;

    case 'E' : longueur=l_ch(ch) ;
                for (i=0 ; i<longueur ; i++)
                    ch[i]=majus(ch[i]) ;
                return(ch) ;

    default : return("") ;
}
}

        /***** chaînes *****/

/* traduction des chiffres romains */
/* retourne l'entier ou 0 sinon */

```

romain(chaine)

```

char chaine[] ;
{
    int nombre[MAXCHAIN+1],          /* tableau de traduction */
        resultat,                    /* valeur finale */
        longueur,                    /* de la chaîne */
        i ;                          /* indentation */

    longueur=l_ch(chaine) ;
    nombre[longueur]=0 ;
    nombre[longueur+1]=0 ;
    for (i=0 ; i<longueur ; i++)
    {
        switch (chaine[i])
        {
            case 'i' : nombre[i]=1 ; break ;
            case 'I' : nombre[i]=1 ; break ;
            case 'v' : nombre[i]=5 ; break ;
            case 'V' : nombre[i]=5 ; break ;
            case 'x' : nombre[i]=10 ; break ;
            case 'X' : nombre[i]=10 ; break ;
            case 'l' : nombre[i]=50 ; break ;
            case 'L' : nombre[i]=50 ; break ;
            case 'c' : nombre[i]=100 ; break ;
            case 'C' : nombre[i]=100 ; break ;
            case 'd' : nombre[i]=500 ; break ;
            case 'D' : nombre[i]=500 ; break ;
            case 'm' : nombre[i]=1000 ; break ;
            case 'M' : nombre[i]=1000 ; break ;
            default : return(0) ;
        }
    }
}

```

```

for (i=0 ; i<(longueur-1) ; i + +)
{
    if (nombre[i] == nombre[i + 1])
    {
        switch (nombre[i])
        {
            case 1 : break ;
            case 10 : break ;
            case 100 : break ;
            case 1000 : break ;
            default : return(0) ;
        }
        if (nombre[i + 2] > nombre[i]) return(0) ;
        if ((nombre[i + 2] == nombre[i]) &&
(nombre[i + 3] == nombre[i]) && (nombre[i] != 1000)) return(0) ;
    }

    if (nombre[i] < nombre[i + 1])
    {
        nombre[i + 1] = nombre[i + 1] - nombre[i] ;
        nombre[i] = 0 ;
        switch (nombre[i + 1])
        {
            case 4 : if (nombre[i + 2] != 0) return(0) ;
                     break ;
            case 9 : if (nombre[i + 2] != 0) return(0) ;
                     break ;
            case 40 : if (nombre[i + 2] > 5) return(0) ;
                     break ;
            case 90 : if (nombre[i + 2] > 5) return(0) ;
                     break ;
            case 400 : if (nombre[i + 2] > 50) return(0) ;
                      break ;
            case 900 : if (nombre[i + 2] > 50) return(0) ;
                      break ;
            default : return(0) ;
        }
    }
}
resultat = nombre[0] ;
for (i = 1 ; i < longueur ; i + +) resultat + = nombre[i] ;
return(resultat) ;
}

```

```

/* test sur les chiffres-> lettres */
/* 7bre, 8bre, 9bre, 10bre, 1er, 1ère */
/* 2nd, 2nde, 2ème... */

```

```
char *test_iem(chiffres, lettres)
```

```

int chiffres ;
char lettres[] ;
{
    extern alphacct() ;

    if ((chiffres > 1) && (!alphacct(lettres, "ème")))
        return("ème") ;
}

```

```

    if ((chiffres == 1) && (!alphacct(lettres, "er")))
        return("premier");
    if ((chiffres == 1) && (!alphacct(lettres, "ère")))
        return("première");
    if ((chiffres == 2) && (!alphacct(lettres, "nd")))
        return("second");
    if ((chiffres == 2) && (!alphacct(lettres, "nde")))
        return("seconde");
    if ((chiffres == 7) && (!alphacct(lettres, "bre")))
        return("septembre");
    if ((chiffres == 8) && (!alphacct(lettres, "bre")))
        return("octobre");
    if ((chiffres == 9) && (!alphacct(lettres, "bre")))
        return("novembre");
    if ((chiffres == 10) && (!alphacct(lettres, "bre")))
        return("décembre");
    return("");
}

        /***** entrées *****/

        /* entrée des données et vérification de leur taille */

char *lit_veri(taillemax, nomfich)

int taillemax;          /* taille maximum des données */
char nomfich[MAXFICH+1]; /* nom du fichier où la modifier */
{
    char tampon[255];      /* tampon de lecture */
    extern void ecrire();

    lit_cl(tampon);
    while (l_ch(tampon) > taillemax)
    {
        printf("\n\nCette entrée est trop longue: la taille maximum autorisée est
%d .", taillemax);
        printf("\n(il est possible de changer cette taille en modifiant sa
définition\nau début du fichier %s)", nomfich);
        ecrire("\n\nEntrez-en une autre: ");
        lit_cl(tampon);
    }
    return(tampon);
}

        /***** fonctions particulières *****/
        /*****/

        /***** menu *****/

```

```

void titre()

```

```

{
    int i, j;                                /* indentation */
    extern void posit_e(), efface_e();

    efface_e();
    for (i=14; i<65; i++) posit_e(" ", 2, i);
    for (i=3; i<22; i++) posit_e(" ", i, 64);
}

```

```

for (i=64 ; i>13 ; i--) posit_e(" ", 22, i) ;
for (i=21 ; i>2 ; i--) posit_e(" ", i, 14) ;
posit_e("RECONNAISSANCE", 6, 32) ;
posit_e("DE SEQUENCES DE MOTS", 9, 29) ;
posit_e("PAR AUTOMATE", 12, 33) ;
posit_e("Adverbes de date du Français", 15, 25) ;
posit_e("Denis MAUREL ", 18, 50) ;

for (j=0 ; j<15 ; j++) for (i=0 ; i<30000 ; i++) ;
efface_e() ;
}

/* menu */
/* retourne le nom du fichier à étudier */

char *menu()
{
    char tampon[80] ;
    int reponse=0,
        i ;
    /* indentation */
    extern void posit_e(), efface_e(), ini_impr(),
        créeindx() ;

    posit_e(": MENU :", 4, 26) ;
    posit_e("Q;;;;;W", 3, 26) ;
    posit_e("A;;;;;S", 5, 26) ;

    posit_e("Voulez-vous:", 10, 1) ;
    posit_e("1.....Avoir une sortie sur imprimante;", 12, 1) ;
    posit_e("2.....Demander la réécriture de l'index du dictionnaire;", 13, 1) ;
    posit_e("3.....Supprimer la sortie sur le fichier de résidus;", 14, 1) ;
    posit_e("4.....Donner le nom de l'article à lire et lancer le programme;", 15, 1) ;
    posit_e("5.....Quitter le programme;", 16, 1) ;
    do
    {
        do
        {
            posit_e(" ", 18, 33) ;
            posit_e("Tapez le numéro de votre choix: ", 18, 1) ;
            lit_cl(tampon) ;
            reponse=car_int(tampon) ;
        }
        while ((reponse!=1)&&(reponse!=2)&&(reponse!=3)&&
            (reponse!=4)&&(reponse!=5)) ;

        if ((reponse>0)&&(reponse<4))
        {
            posit_e("Opération en cours", 20, 1) ;
            for (i=0 ; i<20000 ; i++) ;
        }
        switch (reponse)
        {
            case 1 : ini_impr() ;
                    sort_imp=1 ;
                    break ;

```

```

        case 2 : creeindx() ;
                  break ;

        case 3 : sort_res=0 ;
                  break ;

        case 4 : posit_e("\nQuel est le nom du fichier à lire? ", 20, 1) ;
                  tit_cl(tampon) ;
                  efface_e() ;
                  return(tampon) ;

        case 5 : efface_e ;
                  termine(4) ;
    }
    posit_e("                                ", 20, 1) ;
}
while (reponse!=4) ;
}

```

/***** pointeurs *****/

/* création et initialisation d'un pointeur vers un MOTS */

MOTS *cp_mots(mot, car)

```

char mot[MAXMOT+1], car[MAXPONCT+1] ;
{
    MOTS *pointeur ;
    extern void ecrire() ;

    pointeur=(MOTS*)memoire(taille(MOTS)) ;
    if (!pointeur)
    {
        ecrire("Erreur d'allocation mémoire pour cp_mots!") ;
        return(RIEN) ;
    }
    cop_ch(pointeur->mot, mot) ;
    cop_ch(pointeur->car, car) ;
    pointeur->motdico=RIEN ;
    pointeur->precedent=RIEN ;
    pointeur->suivant=RIEN ;
    pointeur->decale=RIEN ;
    return(pointeur) ;
}

```

/* création et initialisation d'un pointeur vers un INDEX */

INDEX *cp_indx(mot, saut)

```

char mot[MAXMOT+1] ;
int saut ;
{
    INDEX *pointeur ;
    extern void ecrire() ;

    pointeur=(INDEX*)memoire(taille(INDEX)) ;
    if (!pointeur)
    {

```

```

        ecrire("Erreur d'allocation mémoire pour cp_indx!");
        return(RIEN);
    }
    cop_ch(pointeur->mot, mot);
    pointeur->saut=saut;
    pointeur->suivant=RIEN;
    return(pointeur);
}

/* création et initialisation d'un pointeur vers un DICO */

```

DICO *cp_dico(mot, gram, temps)

```

char mot[MAXMOT+1],
      gram[MAXGRAM+1],
      temps[MAXTEMPS+1];
{
    DICO *pointeur;
    extern void ecrire();

    pointeur=(DICO*)memoire(taille(DICO));
    if (!pointeur)
    {
        ecrire("Erreur d'allocation mémoire pour cp_dico!");
        return(RIEN);
    }
    cop_ch(pointeur->mot, mot);
    cop_ch(pointeur->gram, gram);
    cop_ch(pointeur->temps, temps);
    pointeur->suivant=RIEN;
    return(pointeur);
}

```

/* création et initialisation d'un pointeur vers un TRANS */

TRANS *cp_tran(temps, etat)

```

char temps[MAXTEMPS+1];
int etat;
{
    TRANS *pointeur;
    extern void ecrire();

    pointeur=(TRANS*)memoire(taille(TRANS));
    if (!pointeur)
    {
        ecrire("Erreur d'allocation mémoire pour cp_tran!");
        return(RIEN);
    }
    cop_ch(pointeur->temps, temps);
    pointeur->etat=etat;
    pointeur->autre=RIEN;
    return(pointeur);
}

```

/* création et initialisation d'un pointeur vers un CHOIX */

CHOIX *cp_choi(rec_dico, etat, etage)

```
DICO *rec_dico ;
int etat,          /* n° de l'état d'arrivée */
    etage ;        /* par rapport aux MOTS */
{
    CHOIX *pointeur ;
    extern void ecrire() ;

    pointeur=(CHOIX*)memoire(taille(CHOIX)) ;
    if (!pointeur)
    {
        ecrire("Erreur d'allocation mémoire pour cp_choi!") ;
        return(RIEN) ;
    }
    pointeur->reconnu=rec_dico ;
    pointeur->etat=etat ;
    pointeur->etage=etage ;
    pointeur->fils=RIEN ;
    pointeur->pere=RIEN ;
    pointeur->droite=RIEN ;
    pointeur->gauche=RIEN ;
    return(pointeur) ;
}
```

/****** lecture *****/

/* lecture d'un "terme" dans un fichier */
 /* c'est à dire d'un "mot" (suite de lettres ou chiffres) */
 /* et des caractères qui suivent */

char *lit_term(fiche)

```
FICHIER *fiche ;          /* fichier à lire */
                          /* à ouvrir et fermer en dehors */
{
    char c ,              /* caractère lu */
        terme[255] ;      /* terme lu */
    int i, j ;            /* indentation */
    static char detrop ;   /* caractère lu en trop */
    extern est_ltnb() ;

    if (detrop == FIN) return("");

                          /* début du fichier */

    if (!detrop)
    {
        for (i=0 ; (!est_ltnb((c=lit_ca_f(fiche))))&&(c!=FIN) ; i++)
            terme[i]=c ;
        if (c == FIN) return("|") ;
        detrop=c ;
        terme[i]='\0' ;
        return(terme) ;
    }
}
```

```

/* suite */

terme[0]=detrop ;
for (i=1 ; (est_ltnb(c=lit_ca_f(fiche)))&&(c!=FIN) ; i++) terme[i]=c ;
if (i>MAXMOT)
{
    terme[i]='\0' ;
    printf("Attention! le mot %s est trop grand! (%d caractères); il faut
changer la définition de MAXMOT dans le fichier DEFINIR.H", terme, i) ;
    termine(5) ;
}
if (c==FIN)
{
    detrop=c ;
    terme[i]='\0' ;
    return(terme) ;
}
terme[i]=c ;
for (j=i+1 ; (!est_ltnb(c=lit_ca_f(fiche)))&&(c!=FIN) ; j++)
    terme[j]=c ;
detrop=c ;
terme[j]='\0' ;
if ((j-i-1)>MAXPONCT)
{
    printf("Attention! les caractères suivant %s sont trop nombreux! (%d
caractères); il faut changer la définition de MAXPONCT dans le fichier DEFINIR.H",
terme, j-i-1) ;
    termine(6) ;
}
return(terme) ;
}

```

```

/* lecture d'un mot dans un fichier */
/* c'est à dire d'une suite de lettres */

```

```

char *lit_mot(fiche)
FICHIER *fiche ;
/* fichier à lire */
/* à ouvrir et fermer en dehors */
{
    char c , /* caractère lu */
    mot[MAXMOT+1] ; /* mot lu */
    int i ; /* indentation */
    extern char majus() ;

    for (i=0 ; (!est_lt(majus(c=lit_ca_f(fiche))))&&(c!=FIN) ; i++) ;
    if (c==FIN) return("");
    mot[0]=c ;
    for (i=1 ; est_lt(majus(c=lit_ca_f(fiche))) ; i++)
        mot[i]=c ;
    mot[i]='\0' ;
    return(mot) ;
}

```

```

/* complète la lecture d'un mot */
/* dans le fichier dictionnaire (lit_mot ci-dessus) */
/* par le décompte du nombre de caractères associés */

```

```
int lit_nbca(mot)
```

```

char mot[MAXMOT+1];          /* mot lu */
{
    char c;                  /* caractère lu */
    int i;                   /* indentation */

    if (!mot[0]) return(-1);
    for (i=0; (c=lit_ca_f(fic_mots))!='\n'; i++);
    return(i+2);             /* un caractère vient de lit_mot */
                                /* un autre de CRLF */
}

```

```

/* lecture d'un "terme" dans un fichier index */
/* c'est à dire d'un "mot" (suite de lettres) */
/* et du nombre qui suit */

```

```
char *lit_indx()
```

```

{
    char c,                  /* caractère lu */
    terme[MAXMOT+MAXCHIF+1]; /* terme lu */
    int i, j;               /* indentation */
    extern est_ltnb();

    if ((c=lit_ca_f(fic_indx)) == FIN) return("");
    terme[0] = c;
    for (i=1; (est_ltnb((c=lit_ca_f(fic_indx)))) || (c=='\n') || (c=='-'); i++)
    terme[i] = c;
    terme[i] = '\0';
    return(terme);
}

```

```

/* teste la fin d'un adverbe */
/* c'est-à-dire un séparateur de plus d'un caractère */
/* retourne 1 dans ce cas, 0 sinon */

```

```
fin_adv(car)
```

```

char car[MAXPONCT]; /* caractères suivant le mot lu */
{
    if (l_ch(car) != 1) return(1);
    return(0);
}

```

```
/* écrit le début (non alphanumérique) du fichier */
```

```
void debut(article)
```

```

FICHIER *article;          /* fichier à étudier */
{
    char tampon[255];       /* terme lu */
    extern char *lit_term();
    extern void ecrire();
}

```

```

cop_ch(tampon, lit term(article)) ;
if (tampon[0] == '|')
{
    ecrire("Attention! le fichier à lire ne contient ni lettres, ni chiffres!");
    termine(7);
}
ecrire(tampon);
}

```

/***** insertion *****/

/* insertion d'un DICO */

DICO *ins_dico(nouveau, premier)

```

DICO *nouveau, *premier ;
{
    DICO *courant, *precedent ;
    extern alphacct();

    precedent = RIEN ;
    courant = premier ;
    while (courant)
    {
        if (alphacct(courant->mot, nouveau->mot) > 0)
        {
            nouveau->suivant = courant ;
            if (!precedent) return(nouveau) ;
            else
            {
                precedent->suivant = nouveau ;
                return(premier) ;
            }
        }
        if (!alphacct(courant->mot, nouveau->mot))
        {
            cop_ch(courant->gram, nouveau->gram) ;
            cop_ch(courant->temps, nouveau->temps) ;
            return(premier) ;
        }
        precedent = courant ;
        courant = courant->suivant ;
    }
    if (precedent)
    {
        precedent->suivant = nouveau ;
        return(premier) ;
    }
    else return(nouveau) ;
}

```

```

        /***** préparation dico *****/

        /* analyse le résultat de test car */
        /* retourne une chaîne */

char *cas_lrch(ptr_mots, test)

MOTS *ptr_mots;          /* MOTS courant */
char test[];              /* résultat de test_car */
{
    int nbletchi,          /* nombre de lettres ou de chiffres */
        longueur,         /* d'une chaîne de caractères */
        nombre,           /* valeur du chiffre romain */
        i, j;             /* indentation */
    char partie[MAXMOT+1], /* fin du mot */
        resultat[MAXMOT+1]; /* valeur de test_iem */
    extern char *test_iem();
    extern romain();

    switch (test[0])
    {
        case 'A': nbletchi=test[1];
                    longueur=l_ch(ptr_mots->mot);
                    for (i=nbletchi; i<longueur; i++)
                        partie[i-nbletchi]=ptr_mots->mot[i];
                    partie[i-nbletchi]='\0';
                    return(test_iem(car_int(ptr_mots->mot), partie));

        case 'B': if((ptr_mots->car[0] == '%') || (ptr_mots->car[0] == -8))
                    return("");

                    if(car_int(ptr_mots->mot) == 0)
                        return("dnumzéro");
                    if(car_int(ptr_mots->mot) == 1)
                        return("dnumun");
                    if(car_int(ptr_mots->mot) == 10)
                        return("dix");
                    if(car_int(ptr_mots->mot) == 20)
                        return("vingt");
                    if(car_int(ptr_mots->mot) == 30)
                        return("trente");
                    if(car_int(ptr_mots->mot) == 40)
                        return("quarante");
                    if(car_int(ptr_mots->mot) == 50)
                        return("cinquante");
                    if(car_int(ptr_mots->mot) == 60)
                        return("soixante");
                    if((car_int(ptr_mots->mot) == 70) ||
(car_int(ptr_mots->mot) == 80) || (car_int(ptr_mots->mot) == 90))
                        return("dnumdecennie");
                    if((car_int(ptr_mots->mot) > 1) && (car_int(ptr_mots-
> mot) < 13))
                        return("dnummois");
                    if((car_int(ptr_mots-
> mot) > 12) && (car_int(ptr_mots-> mot) < 25))
                        return("dnumheure");
                    if((car_int(ptr_mots-
> mot) > 24) && (car_int(ptr_mots-> mot) < 32))
                        return("dnumjour");
    }
}

```

```

        if((car_int(ptr_mots-
> mot)>31)&&(car_int(ptr_mots->mot)<61))
            return("dnumminute");
        if((car_int(ptr_mots-
> mot)>60)&&(car_int(ptr_mots->mot)<99))
            return("dnuman");
        if((car_int(ptr_mots-
> mot)>99)&&(car_int(ptr_mots->mot)<9999))
            return("dnumannee");
        return("dnum");

    case 'C' : return(ptr_mots->mot);

    case 'D' : nbletchi=test[1];
                longueur=l_ch(ptr_mots->mot);

                /* chiffres romains */

                for (i=0 ; i<nbletchi ; i++)
                    partie[i]=ptr_mots->mot[i];
                partie[i]='\0';
                nombre=romain(partie);
                if (nombre)
                {
                    for (i=nbletchi ; i<longueur ; i++) partie[i-
nbletchi]=ptr_mots->mot[i];

                    partie[i-nbletchi]='\0';
                    cop_ch(resultat, test_iem(nombre, partie));
                    if((nbletchi!=1) || (resultat[0]))
                        return(resultat);
                }

                /* autres */

                if (nbletchi!=1) return("");
                return(ptr_mots->mot);

    case 'E' : longueur=l_ch(ptr_mots->mot);

                /* chiffres romains */

                nombre=romain(ptr_mots->mot);
                if ((nombre)&&(ptr_mots->car[0]==' '))
                {
                    if (nombre==1) return("rmi");
                    if (nombre<13) return("rmmois");
                    if (nombre<32) return("rmjour");
                    if (nombre==101) return("rmci");
                    if (nombre==509) return("rmdix");
                    if (nombre==1001) return("rmmi");
                    if (nombre<9999) return("rman");
                    return("rm");
                }

                /* autres */

                return(ptr_mots->mot);

    default :    return("");
}

```

```

}

        /****** consultation *****/

        /* charge les noms des fichiers utilisés: */
        /* index (1), dictionnaire (2), */
        /* automate (3) et résidus (4) */

char *chargnom(numero)

int numero ;          /* du nom à charger */
{
    int slash,          /* compte les "/" */
        i,              /* indentation */
        c ;             /* caractère lu */
    char nom[MAXFICH+1] ;
    FICHIER *fiche ;    /* fichier des noms de fichier */
    extern void ecrire() ;

        /* ouverture du fichier */

    fiche = ouv_1_f("ADVDATE.NOM") ;
    if (!fiche)
    {
        ecrire("\nIl y a erreur: le fichier ADVDATE.NOM n'est pas ici") ;
        termine(8) ;
    }

    slash = 0 ;
    while ((c = lit_ca_f(fiche)) != FIN)
    {
        nom[0] = c ;
        for (i = 1 ; (c = lit_ca_f(fiche)) != '/' ; i++) nom[i] = c ;
        nom[i] = '\0' ;
        if (++slash == numero) break ;
    }

        /* fermeture du fichier */

    if ((ferme_f(fiche)))
        ecrire("\nAttention! erreur dans la fermeture du fichier ADVDATE.NOM!") ;

    return(nom) ;
}

        /* chargement du dictionnaire à partir d'un mot de l'index */

DICO *chargdic(ptr_idx)

INDEX *ptr_idx ;
{
    char mot[MAXMOT+1],          /* mot à lire */
        gram[MAXGRAM+1],        /* informations grammaticales */
        temps[MAXTEMPS+1],      /* informations sur les états */
        c ;                      /* caractère lu */
    int i, j ;                   /* indentation */
    long saut ;                  /* déplacement dans le fichier */
    DICO *pre_dico, *cou_dico, *der_dico ;
    extern alphacct() ;

```

```

extern char majus();
extern void ecrire();
extern DICO *cp_dico();

/* lecture du premier terme */

saut=ptr_indx->saut;
if (pos_débf(fic_mots, saut)!=0)
{
    ecrire("Attention ! erreur de positionnement dans le fichier!");
    termine(9);
}

/* exemple: au/(à le Pré#PRLE)(à le Pré#PR) */

for (i=0; (c=lit_ca_f(fic_mots))!='/' ; i++) mot[i]=c;
mot[i]='\0'; /* au/ */
c=lit_ca_f(fic_mots); /* ( */
for (i=0; (c=lit_ca_f(fic_mots))!='#' ; i++) gram[i]=c;
gram[i]='\0'; /* à le Pré# */
for (i=0; (c=lit_ca_f(fic_mots))!=')' ; i++) temps[i]=c;
temps[i]='\0'; /* PRLE) */
pre_dico=cp_dico(mot, gram, temps);
cou_dico=pre_dico;
while ((c=lit_ca_f(fic_mots))== '(') /* ( /* */
{
    for (i=0; (c=lit_ca_f(fic_mots))!='#' ; i++)
        gram[i]=c;
    gram[i]='\0'; /* à le Pré# */
    for (i=0; (c=lit_ca_f(fic_mots))!=')' ; i++)
        temps[i]=c;
    temps[i]='\0'; /* PR) */
    der_dico=cp_dico(mot, gram, temps);
    cou_dico->suivant=der_dico;
    cou_dico=der_dico;
}
if ((c=lit_ca_f(fic_mots))!=FIN);
{
    mot[0]=c;
    for (i=1; est_lt(majus((c=lit_ca_f(fic_mots)))) ; i++)
        mot[i]=c;
    mot[i]='\0';
    /* puis des suivants */

    while ((!ptr_indx->suivant) || ((alphacct(mot, (ptr_indx->suivant)-
>mot))))
    {
        if (c!='/')
        {
            mot[i]=c;
            for (j=i+1; (c=lit_ca_f(fic_mots))!='/' ; j++)
                mot[j]=c;
            mot[j]='\0';
        }
        while ((c=lit_ca_f(fic_mots))== '(')
        {
            for (i=0; (c=lit_ca_f(fic_mots))!='#' ; i++)
                gram[i]=c;
            gram[i]='\0';

```

```

        for (i=0 ; (c=lit_ca_f(fic_mots))!='' ; i++)
            temps[i]=c ;
        temps[i]='\0' ;
        der_dico=cp_dico(mot, gram, temps) ;
        cou_dico->suivant=der_dico ;
        cou_dico=der_dico ;
    }
    if ((c=lit_ca_f(fic_mots))==FIN) break ;
    mot[0]=c ;
    for (i=1 ; est_l(majus((c=lit_ca_f(fic_mots)))) ; i++)
        mot[i]=c ;
    mot[i]='\0' ;
}
}
return(pre_dico) ;
}

```

```

/* recherche d'un mot dans le dictionnaire */
/* retourne la liste correspondant à l'index */
/* ou RIEN si la chaîne est vide */
/* ou inférieure au premier mot de l'index */

```

DICO *chrchdic(motdudic)

```

char motdudic[] ;          /* mot à chercher */
{
    char terme[MAXMOT+MAXCHIF+2], /* terme lu */
          mot[2][MAXMOT+1],        /* mot lu */
          chiffres[2][MAXCHIF+1] ; /* chiffres lus */
    int i, j,                      /* indentation */
        longueur ;                /* d'une chaîne de caractères */
    INDEX *pre_indx ;
    DICO *pre_dico ;
    extern DICO *chargdic() ;
    extern INDEX *cp_indx() ;
    extern alpha() ;
    extern char majus(), *lit_indx() ;

    if (!motdudic[0]) return(RIEN) ;

    /* lecture de l'index du dictionnaire */

    ini_f(fic_indx) ;

    /* premier terme */

    cop_ch(terme, lit_indx()) ;
    if (terme[0])
    {
        longueur=l_ch(terme) ;
        for (i=0 ; !est_nb(majus(terme[i])) ; i++)
            mot[0][i]=terme[i] ;
        mot[0][i]='\0' ;
        for (j=0 ; i<longueur ; j++, i++)
            chiffres[0][j]=terme[i] ;
        chiffres[0][j]='\0' ;
    }
    if (alpha(motdudic, mot[0])<0) pre_dico=RIEN ;
}

```

```

else                                     /* termes suivants */
{
    cop_ch(terme, lit_indx());
    while (terme[0])
    {
        longueur=1; ch(terme);
        for (i=0; !est_nb(majus(terme[i])); i++)
            mot[1][i]=terme[i];
        mot[1][i]='\0';
        for(j=0; i<longueur; j++, i++)
            chiffres[1][j]=terme[i];
        chiffres[1][j]='\0';
        if (alpha(motdudic, mot[1])<0) break;
        cop_ch(mot[0], mot[1]);
        cop_ch(chiffres[0], chiffres[1]);
        cop_ch(terme, lit_indx());
    }

    /* utilisation d'une liste chaînée de deux termes */
    /* encadrant le mot à chercher */

    pre_indx=cp_indx(mot[0], car_int(chiffres[0]));
    if (terme[0]) pre_indx->suivant=cp_indx(mot[1], car_int(chiffres[1]));
    pre_dico=chargdic(pre_indx);
    if (terme[0]) libere(pre_indx->suivant);
    libere(pre_indx);
}
return(pre_dico);
}

/* égalité des mots composés */
/* retourne le nombre de mots simples (decalage) si égalité */
/* 0 sinon */

```

mot_comp(ptr_mots, motdudic)

```

MOTS *ptr_mots;                        /* mot à comparer au */
char motdudic[MAXMOT+1];              /* mot du dictionnaire */
{
    char decoupe[MAXPART][MAXMOT+1], /* découpage en */
        separe[MAXPART+1],           /* séparateurs */
        format,                       /* pour min_maj */
        c;                            /* caractère étudié */
    int i, j, k,                       /* indentation */
        test;                         /* test d'égalité */
    MOTS *cou_mots;
    DICO *cou_dico;
    extern char *test_car(), *min_maj(), majus();
    extern alphacct();

    /* découpage du mot du dictionnaire */

    c=motdudic[0];
    for (i=0, j=0, k=0; i<MAXPART; i++, j=0)
    {
        while (est_lt(majus(c)))
        {

```

```

        decoupe[i][j]=c ;
        j++ ; k++ ;
        c=motdudic[k] ;
    }
    decoupe[i][j]='\0' ;

    if (c=='\0') break ;
    separe[i]=c ;
    c=motdudic[+ + k] ;
}

/* comparaison */

cou_mots=ptr_mots ;

for (j=0 ; j<i+1 ; j++)
{
    format=test_car(cou_mots->mot)[0] ;
    cou_dico=cou_mots->motdico ;
    test=alphacct(min_maj(cou_dico->mot, format), min_maj(decoupe[j],
format)) ;
    if (j==i)
    {
        if (!test) return(i+1) ;
        else return(0) ;
    }
    if (test || (l_ch(cou_mots->car)!=1) || (cou_mots->car[0]!=separe[j]))
return(0) ;
    cou_mots=cou_mots->suivant ;
}
}

```

/***** automates *****/

/* examine la liste de MOTS sortie de cl mots et */
/* cherche des mots-clefs pour tester l'utilité de */
/* la comparaison avec l'automate; retourne 0 sinon */

testauto(ptr_mots)

```

MOTS *ptr_mots ;
{
    int i ;
    MOTS *cou_mots ;
    DICO *cou_dico ;

/* indentation */

/* tests */

if (!ptr_mots) return(0) ;
cou_mots=ptr_mots ;
while (cou_mots)
{
    cou_dico=cou_mots->motdico ;
    while (cou_dico)
    {
        /* Advtps */
        if (!comp_ch(cou_dico->temps,"AD")) return(1) ;
    }
}

```

```

/* Nfête */
if (!comp_ch(cou_dico->temps,"NF")) return(1) ;

/* Ntps */
if (!comp_ch(cou_dico->temps,"NT")) return(1) ;

/* Date */
if (((!comp_ch(cou_dico->temps,"DJ"))||(!comp_ch(cou_dico-
>temps,"NJ"))||(!comp_ch(cou_dico->temps,"NM"))||(!comp_ch(cou_dico-
>temps,"DN")))) return(1) ;

/* Heure */
if (((!comp_ch(cou_dico->temps,"HE"))||(!comp_ch(cou_dico-
>temps,"MI"))||(!comp_ch(cou_dico->temps,"QA"))||(!comp_ch(cou_dico-
>temps,"QT"))||(!comp_ch(cou_dico->temps,"DI")))) return(1) ;

    cou_dico=cou_dico->suivant ;
}
cou_mots=cou_mots->suivant ;
}
return(0) ;
}

```

```

/* lit l'automate et remplit un tableau de pointeurs */
/* de taille MAXETATS */

```

```
void ct_tran(automate)
```

```

TRANS *automate[MAXETATS+1] ;          /* tableau des états */
{
    int final[MAXETATS-1], /* test si un état est final */
        /* 0 si l'état n'est pas final, 1 sinon */
        etat, /* numero de l'état suivant */
        /* négatif si cet état est final */
        c, /* caractère lu */
        i,j ; /* indentation */
    char temps[MAXTEMPS+1],
        numero[3] ; /* de l'état */
    TRANS *cou_tran, *nou_tran ;
    extern TRANS *cp_tran() ;

    ini_f(fic_tran) ;

        /* lecture des états */

    for (i=0 ; (c=lit_ca_f(fic_tran))!=FIN ; i++)
    {
        /* première transition */

        if (i) final[i-1]=car_int(&c) ;
        for (j=0 ; est_lt((c=lit_ca_f(fic_tran))) ; j++)
            temps[j]=c ;
        temps[j]='\0' ;
        numero[0]=c ;
        if (est_nb((c=lit_ca_f(fic_tran))))
        {
            numero[1]=c ;
            numero[2]='\0' ;

```

```

        c=lit_ca_f(fic_tran);
    }
    else numero[1]='\0';
    etat=car_int(numero);
    automate[i]=cp_tran(temps, etat);
    cou_tran=automate[i];

        /* autres transitions éventuelles */

while (c!='/')
{
    for (j=0; est_lt((c=lit_ca_f(fic_tran))); j++)
        temps[j]=c;
    temps[j]='\0';
    numero[0]=c;
    if (est_nb((c=lit_ca_f(fic_tran))))
    {
        numero[1]=c;
        numero[2]='\0';
        c=lit_ca_f(fic_tran);
    }
    else numero[1]='\0';
    etat=car_int(numero);
    nou_tran=cp_tran(temps, etat);
    cou_tran->autre=nou_tran;
    cou_tran=nou_tran;
}

        /* gestion des états finaux */

for (j=0; j<i; j++)
{
    cou_tran=automate[j];
    if (final[(cou_tran->etat)-1])
        cou_tran->etat=-(cou_tran->etat);
    nou_tran=cou_tran->autre;
    while (nou_tran)
    {
        if (final[(nou_tran->etat)-1])
            nou_tran->etat=-(nou_tran->etat);
        nou_tran=nou_tran->autre;
    }
}

/* compare avec l'automate la chaîne de MOTS */
/* retourne l'arbre des CHOIX possibles */
/* ou RIEN si aucun MOTS n'est reconnu */

CHOIX *clf_choi(pre_mots)

MOTS *pre_mots;                                /* premier MOTS à étudier */
{
    int etat,
        i;                                        /* indentation */
    MOTS *cou_mots;
    TRANS *automate[MAXETATS+1];                /* tableau des états */
    CHOIX *pre_choi, *cou_choi, *der_choi;

```

```

extern void ct_tran(), libauto();
extern CHOIX *cla_choi(), *droite();

/* lecture de l'automate */

if (!pre_mots) return(RIEN);
ct_tran(automate);

/* premier MOTS */

pre_choi=cla_choi(automate, pre_mots, 0, 1);
if (!pre_choi)
{
    libauto(automate);
    return(RIEN);
}

/* MOTS suivants */

cou_mots=pre_mots;
der_choi=pre_choi;
for (i=1; (cou_mots=cou_mots->decale); i++)
{
    der_choi=droite(pre_choi);
    while (der_choi)
    {
        if (der_choi->etage == i)
            cou_choi=cla_choi(automate, cou_mots, der_choi->etat,
i+1);
        else cou_choi=RIEN;
        if (cou_choi)
        {
            der_choi->fils=cou_choi;
            cou_choi->pere=der_choi;
        }
        if (der_choi->gauche)
        {
            der_choi=der_choi->gauche;
            if (der_choi->fils)
                der_choi=droite(der_choi->fils);
            continue;
        }
        while (der_choi->pere) der_choi=der_choi->pere;
        if (!der_choi->gauche) break;
        der_choi=der_choi->gauche;
        if (der_choi->fils)
            der_choi=droite(der_choi->fils);
        else der_choi=RIEN;
    }
}
libauto(automate);
return(pre_choi);
}

```

```
/* libère l'automate */
```

```
void libauto(automate)
```

```
TRANS *automate[MAXETATS + 1];      /* tableau des états */
{
    int i;                          /* indentation */
    TRANS *cou_tran;

    for (i=0; i<MAXETATS; i++)
    {
        cou_tran=automate[i];
        do libere(cou_tran);
        while ((cou_tran=cou_tran->autre));
    }
}
```

```
/* à partir d'un MOTS et d'un état */
/* retourne le CHOIX de transitions reconnues */
/* ou RIEN */
```

```
CHOIX *cla_choi(automate, ptr_mots, etat, etage)
```

```
TRANS *automate[MAXETATS + 1];      /* tableau des états */
MOTS *ptr_mots;                     /* MOTS à étudier */
int etat,                           /* n° de l'état d'arrivée */
    etage;                           /* par rapport aux MOTS */
{
    int suite1, suite2;              /* n° d'un état */
    TRANS *cou_tran, *nou_tran;
    DICO *cou_dico;
    CHOIX *pre_choi, *cou_choi, *der_choi;
    extern CHOIX *cp_choi();

    pre_choi=RIEN;
    suite1=etat;
    if (suite1<0) suite1=-suite1;
    cou_dico=ptr_mots->motdico;

    while (cou_dico)
    {
        cou_tran=automate[suite1];
        while (cou_tran)
        {
            if ((l_ch(cou_dico->temps)==2)&&(!comp_ch(cou_tran->temps,
cou_dico->temps)))
            {
                der_choi=cp_choi(cou_dico, cou_tran->etat, etage);
                if (!pre_choi) pre_choi=der_choi;
                else
                {
                    cou_choi->droite=der_choi;
                    der_choi->gauche=cou_choi;
                }
                cou_choi=der_choi;
            }
        }
    }
}
```

```

        if ((1_ch(cou_dico->temps)==4)&&((cou_tran-
>temps)[0]==(cou_dico->temps)[0])&&((cou_tran->temps)[1]==(cou_dico-
>temps)[1]))
        {
            suite2=cou_tran->etat ;
            if (suite2<0) suite2=-suite2 ;
            nou_tran=automate[suite2] ;
            while (nou_tran)
            {
                if (((nou_tran->temps)[0]==(cou_dico-
>temps)[2])&&((nou_tran->temps)[1]==(cou_dico->temps)[3]))
                {
                    der_choi=cp_choi(cou_dico, nou_tran->etat,
etage) ;
                    if (!pre_choi)
                        pre_choi=der_choi ;
                    else
                    {
                        cou_choi->droite=der_choi ;
                        der_choi->gauche=cou_choi ;
                    }
                    cou_choi=der_choi ;
                }
                nou_tran=nou_tran->autre ;
            }
            cou_tran=cou_tran->autre ;
        }
        cou_dico=cou_dico->suivant ;
    }
    return(pre_choi) ;
}

```

/* se place le plus à droite dans l'arbre des CHOIX */

CHOIX *droite(ptr_choi)

```

CHOIX *ptr_choi ;
{
    CHOIX *cou_choi, *der_choi ;

    cou_choi=ptr_choi ;
    while (cou_choi)
    {
        while (cou_choi->droite) cou_choi=cou_choi->droite ;
        der_choi=cou_choi ;
        cou_choi=cou_choi->fils ;
    }
    return(der_choi) ;
}

```

```

/* compte le nombre de noeuds jusqu'à la racine */
/* de l'arbre des CHOIX */
/* ajoute 1 pour obtenir le nombre de MOTS reconnus */
/* retourne 0 si la feuille ne correspond pas */
/* à un état final */

```

```
comptnoe(ptr_choi, pre_choi)
```

```

CHOIX *ptr_choi,          /* feuille de départ */
      *pre_choi;          /* racine */
{
    int nombre=1;         /* de noeuds */
    CHOIX *cou_choi;

    cou_choi=ptr_choi;
    if (cou_choi->etat>0) return(0);

    while(cou_choi!=pre_choi)
    {
        while (cou_choi->gauche) cou_choi=cou_choi->gauche;
        while (cou_choi->pere)
        {
            nombre++ ;
            cou_choi=cou_choi->pere;
        }
    }
    return(nombre);
}

```

```

/* libère les DICO d'un MOTS, un seul excepté */
/* (teste le champ "temps") */

```

```
MOTS *lib_dico(ptr_mots, temps)
```

```

MOTS *ptr_mots;          /* MOTS à "alléger" */
char temps[MAXTEMPS+1];  /* DICO à conserver */
{
    DICO *cou_dico;

    cou_dico=ptr_mots->motdico;

    while (comp_ch(cou_dico->temps, temps))
    {
        ptr_mots->motdico=cou_dico->suivant;
        libere(cou_dico);
        cou_dico=ptr_mots->motdico;
    }

    while ((cou_dico=cou_dico->suivant)) libere(cou_dico);

    (ptr_mots->motdico)->suivant=RIEN;
    return(ptr_mots);
}

```

```

/* modifie le champ DICO */
/* lors du passage entre deux adverbes */

```

MOTS *passage(ptr_mots, temps)

```

MOTS *ptr_mots ;          /* mot du passage */
char temps[] ;            /* champ temps à conserver */
{
    char mot[MAXMOT+1],
          suite[MAXTEMPS+1] ;
    MOTS *cou_mots ;
    DICO *cou_dico ;
    extern MOTS *lib_dico() ;

    lib_dico(ptr_mots, temps) ;
    cou_dico = ptr_mots->motdico ;
    cop_ch(mot, "&") ;
    cat_ch(mot, cou_dico->mot) ;
    cop_ch(cou_dico->mot, mot) ;
    if (l_ch(temps) == 4)
    {
        cou_dico->temps[2] = '\0' ;
        cou_dico = (ptr_mots->decale)->motdico ;
        while (cou_dico)
        {
            suite[0] = temps[2] ;
            suite[1] = temps[3] ;
            suite[2] = '\0' ;
            cat_ch(suite, cou_dico->temps) ;
            cop_ch(cou_dico->temps, suite) ;
            cou_dico = cou_dico->suivant ;
        }
    }
    return(ptr_mots) ;
}

```

```

/***** tables de transitions *****/

```

```

/* lecture de l'en-tête d'une table */
/* retourne la colonne correspondante ou 0 */

```

lire_ent(temps, mot)

```

char temps[MAXTEMPS+1],   /* nom de la table */
      mot[MAXMOT+1] ;     /* mot à chercher */
{
    char nomfich[MAXFICH+1], /* nom du fichier */
          entete[MAXMOT+1],  /* mot lu */
          c ;                /* caractère lu */
    int colonne = 1,         /* numero de la colonne */
        trouve = 0,         /* test */
        i ;                 /* indentation */
    FICHIER *fiche ;        /* fichier d'en-tête */
    extern alphacct() ;

    /* chargement du fichier */

    cop_ch(nomfich, temps) ;

```

```

cat ch(nomfich, ".ENT");
fiche = ouv_l_f(nomfich);
if (!fiche)
{
    printf("\nIl y a erreur dans l'ouverture du fichier %s!", nomfich);
    termine(10);
}

/* lecture du fichier */

while ((c = lit_ca_f(fiche)) != FIN)
{
    entete[0] = c;
    for (i = 1; ((c = lit_ca_f(fiche)) != ' ') && (c != '/'); i++) entete[i] = c;
    entete[i] = '\0';

    if (!alphacct(mot, entete))
    {
        trouve = 1;
        break;
    }
    if (c == '/') colonne++;
}

/* fermeture du fichier */

if ((ferme_f(fiche)))
    printf("\nAttention! erreur dans la fermeture du fichier %s!", nomfich);

if (trouve) return(colonne);
return(0);
}

/* lecture du corps d'une table */
/* retourne 0 si le fichier n'existe pas, 1 si +, */
/* -1 si -, -2 si le mot n'est pas trouvé */
/* -3 si la colonne est nulle (en-tête non trouvée) */

lire_tab(mot, nomfich, colonne)

char mot[MAXMOT+1], /* mot à chercher */
nomfich[MAXFICH+1]; /* nom sans extension */
int colonne; /* numero de la colonne */
{
    char corps[MAXMOT+1], /* mot lu */
nouvnom[MAXFICH+1], /* nom avec extension */
c; /* caractère lu */
    int trouve = 0, /* test */
i; /* indentation */
    FICHIER *fiche; /* fichier .TAB */
    extern alphacct();

    if (!colonne) return(-3);

    /* chargement du fichier */

    cop_ch(nouvnom, nomfich);
    cat_ch(nouvnom, ".TAB");
    fiche = ouv_l_f(nouvnom);
    if (!fiche) return(0);

```

```

/* lecture du fichier */

while ((c=lit_ca_f(fiche))!=FIN)
{
    corps[0]=c ;
    for (i=1 ; ((c=lit_ca_f(fiche))!='+')&&(c!=':'); i++) corps[i]=c ;
    corps[i]='\0' ;

    if (!alphacct(mot, corps))
    {
        trouve=1 ;
        break ;
    }
    if (c==':')
        for (i=1 ; (c=lit_ca_f(fiche))!='/' ; i++) ;
}

if (trouve)
{
    while (c!=':') c=lit_ca_f(fiche) ;
    for (i=0 ; i<colonne ; i++) c=lit_ca_f(fiche) ;
}

/* fermeture du fichier */

if ((ferme f(fiche)))
    printf("\nAttention! erreur dans la fermeture du fichier %s!", nouvnom) ;

if (!trouve) return(-2) ;
if (c=='+') return(1) ;
if (c==':') return(-1) ;
printf("\nVérifier le fichier %s", nouvnom) ;
return(0) ;
}

```

```

/* contracte les dates de lst_tps en un seul symbole: CA */
/* met le tableau lst_mots à jour */
/* marque chaque terme de la liste par @ */

```

```

char *contr_ca(lst_tps, lst_mots)

```

```

char lst_tps[MAXLIST*2+1] ; /* chaîne des champs temps */
MOTS *lst_mots[MAXLIST] ; /* tableau de pointeurs */
/* vers les MOTS de la liste */
{
    int longueur, /* de la liste */
        i ; /* indentation */
    char chaine[5], /* partie de chaîne à comparer */
        mot[MAXMOT+1], /* pour marquer les dates */
        avant[MAXTEMPS+1] ; /* temps d'avant */
    DICO *cou dico ;
    extern char *part_ch(), *contrali() ;

    longueur=l_ch(lst_tps) ;
    for (i=0 ; i<longueur/2 ; i++)
    {
        cop_ch(chaine, part_ch(lst_tps, 2*i, 2*i+1)) ;
    }
}

```

```

if ((!(comp_ch(chaine, "DA")))| |
    (!(comp_ch(chaine, "DB")))| |
    (!(comp_ch(chaine, "DH")))| |
    (!(comp_ch(chaine, "DI")))| |
    (!(comp_ch(chaine, "DJ")))| |
    (!(comp_ch(chaine, "DK")))| |
    (!(comp_ch(chaine, "DM")))| |
    (!(comp_ch(chaine, "HE")))| |
    (!(comp_ch(chaine, "MI")))| |
    (!(comp_ch(chaine, "MN")))| |
    (!(comp_ch(chaine, "MS")))| |
    (!(comp_ch(chaine, "NJ")))| |
    (!(comp_ch(chaine, "NM")))| |
    (!(comp_ch(chaine, "QA")))| |
    (!(comp_ch(chaine, "QT")))| |
    (!(comp_ch(chaine, "SE"))))
{
    lst_tps[2*i]='C';
    lst_tps[2*i+1]='A';
    cou_dico=lst_mots[i]->motdico;
    cop_ch(mot, "@");
    cat_ch(mot, cou_dico->mot);
    cop_ch(cou_dico->mot, mot);
}
if ((!(comp_ch(chaine, "HE")))| |
    (!(comp_ch(chaine, "MN")))| |
    (!(comp_ch(chaine, "SE"))))
{
    cop_ch(avant, part_ch(lst_tps, 2*i-2, 2*i-1));
    if ((i)&&(!(comp_ch(avant, "DN"))))
        cop_ch(lst_tps, contrali(lst_tps, lst_mots, --i));
}
if ((!(comp_ch(chaine, "DJ")))| |
    (!(comp_ch(chaine, "NJ"))))
{
    cop_ch(avant, part_ch(lst_tps, 2*i-2, 2*i-1));
    if ((i)&&(!(comp_ch(avant, "LE")))| |
        (!(comp_ch(avant, "CE")))| |(!(comp_ch(avant, "UN")))))
        cop_ch(lst_tps, contrali(lst_tps, lst_mots, --i));
}
if ((!(comp_ch(chaine, "DI")))| |
    (!(comp_ch(chaine, "QA"))))
{
    cop_ch(avant, part_ch(lst_tps, 2*i-2, 2*i-1));
    if ((i)&&(!(comp_ch(avant, "LE"))))
        cop_ch(lst_tps, contrali(lst_tps, lst_mots, --i));
}
}

for (i=0 ; i<longueur/2 ; i++)
{
    cop_ch(chaine, part_ch(lst_tps, 2*i, 2*i+3));
    if (!(comp_ch(chaine, "CACA")))
    {
        cop_ch(lst_tps, contrali(lst_tps, lst_mots, i+1));
        i--;
    }
}

```

```

    return(lst_tps) ;
}

/* contracte une partie des deux listes */

char *contrali(lst_tps, lst_mots, i)

char lst_tps[MAXLIST*2+1] ; /* chaîne des champs temps */
MOTS *lst_mots[MAXLIST] ; /* tableau de pointeurs */
int i ; /* à partir de i */
{
    int longueur, /* de la liste */
        j ; /* indentation */

    longueur=l_ch(lst_tps) ;
    if (i==longueur) lst_tps[2*i]='\0' ;

    for (j=i ; j<longueur/2 ; j++)
    {
        lst_mots[j]=lst_mots[j+1] ;
        lst_tps[2*j]=lst_tps[2*j+2] ;
        lst_tps[2*j+1]=lst_tps[2*j+3] ;
    }
    lst_tps[2*j]='\0' ;
    return(lst_tps) ;
}

/* écrase une partie des deux listes */
/* et la remplace par le symbole $$ */

char *ecralist(lst_tps, lst_mots, i, pre_mots)

char lst_tps[MAXLIST*2+1] ; /* chaîne des champs temps */
MOTS *lst_mots[MAXLIST] ; /* tableau de pointeurs */
int i ; /* à partir de i */
MOTS *pre_mots ;
{
    int longueur, /* de la liste */
        j, k, l ; /* indentation */
    MOTS *cou_mots, *j_mots ;
    DICO *cou_dico ;
    extern char *contrali() ;

    longueur=l_ch(lst_tps)/2 ;
    for (j=i ; j>0 ; j--)
    {
        cou_mots=lst_mots[j] ;
        cou_dico=cou_mots->motdico ;
        while ((cou_dico->mot[0]!='&')&&(cou_dico-
> mot[1]!='&')&&(cou_mots!=lst_mots[j-1]))
        {
            cou_mots=cou_mots->precedent ;
            cou_dico=cou_mots->motdico ;
        }
        if ((cou_dico->mot[0]=='&')|| (cou_dico->mot[1]=='&'))
            break ;
    }
    if (!j) j_mots=pre_mots ;
}

```

```

else j_mots=cou_mots->decale ;

for (k=i ; k<longueur ; k++ )
{
    cou_mots=lst_mots[k] ;
    cou_dico=cou_mots->motdico ;
    while (((cou_dico->mot[0]!='&') || (cou_dico-
> mot[1]!='&'))&&(cou_mots!=lst_mots[k+1])&&(cou_mots))
    {
        cou_mots=cou_mots->suivant ;
        cou_dico=cou_mots->motdico ;
    }
    if ((cou_dico->mot[0]=='&') || (cou_dico->mot[1]=='&'))
        break ;
}

if (k==longueur) lst_tps[2*j+2]='\0' ;
else for (l=j ; l<k ; l++)
    cop_ch(lst_tps, contrali(lst_tps, lst_mots, j)) ;

lst_tps[2*j]='$' ;
lst_tps[2*j+1]='$' ;
lst_mots[j]=j_mots ;

return(lst_tps) ;
}

```

/* contracte les modificateurs de la liste en les supprimant */

char *contr_mo(lst_tps, lst_mots, pre_mots)

```

char lst_tps[MAXLIST*2+1] ; /* chaîne des champs temps */
MOTS *lst_mots[MAXLIST], /* tableau de pointeurs */
    *pre_mots ; /* vers les MOTS de la liste */
{
    int longueur, /* de la liste */
        test, /* l'acceptation des formes étudiées */
        colonne, /* de la table */
        i, j ; /* indentation */
    char chaine[5], /* partie de chaîne à comparer */
        date[5], /* premier temps de CA */
        avant[5], /* temps d'avant */
        temps[MAXTEMPS+1], /* table d'en-tête */
        entete[MAXMOT+1], /* mots à */
        corps[MAXMOT+1] ; /* chercher */
    MOTS *cou_mots ;
    DICO *cou_dico ;
    extern char *part_ch(), *contrali(), *ecralist(),
        *entetcor() ;
    extern lire_tab(), lire_ent() ;

    longueur=l_ch(lst_tps) ;
    for (i=0 ; i<longueur/2 ; i++)
    {
        test=0 ;
        cop_ch(chaine, part_ch(lst_tps, 2*i, 2*i+3)) ;
    }
}

```

```

if ((!(comp_ch(chaine, "MONF")))|
    (!(comp_ch(chaine, "MONT")))|
    (!(comp_ch(chaine, "NFMO")))|
    (!(comp_ch(chaine, "NTMO")))|
    (!(comp_ch(chaine, "MOCA")))|
    (!(comp_ch(chaine, "CAMO"))))
{
    /* sans date */

    if ((!(comp_ch(chaine, "MONF")))|
        (!(comp_ch(chaine, "MONT")))|
        (!(comp_ch(chaine, "NFMO")))|
        (!(comp_ch(chaine, "NTMO"))))
    {
        cop_ch(temps, part_ch(lst_tps, 2*i+2, 2*i+3));
        cop_ch(entete, entetcor(lst_mots[i+1]));
        colonne=lire_ent(temps, entete);
        cop_ch(corps, entetcor(lst_mots[i]));
        test=lire_tab(corps, chaine, colonne);
        if (test>0) /* on contracte MO */
            for (j=i; j<i+2; j++)
            {
                cop_ch(chaine, part_ch(lst_tps, 2*j, 2*j+1));
                if (!(comp_ch(chaine, "MO")))
                {
                    cop_ch(lst_tps, contrali(lst_tps,
lst_mots, j));
                    i--;
                }
            }
        }

        /* avec date */

        if (!(comp_ch(chaine, "MOCA")))
        {
            cop_ch(temps, "CA");
            cop_ch(avant, part_ch(lst_tps, 2*i-2, 2*i-1));
            if ((i)&&(!(comp_ch(avant, "LE")))|
                (!(comp_ch(avant, "CE")))|
                (!(comp_ch(avant, "UN")))|
                (!(comp_ch(avant, "DN")))))
                cop_ch(entete, avant);
            else cop_ch(entete, "");
            cou_mots=lst_mots[i+1];
            cou_dico=cou_mots->motdico;
            do
            {
                cat_ch(entete, cou_dico->temps);
                cou_mots=cou_mots->decale;
                cou_dico=cou_mots->motdico;
            }
            while ((cou_mots)&&(cou_dico->mot[0]='@'));
            colonne=lire_ent(temps, entete);
            cop_ch(corps, entetcor(lst_mots[i]));
            test=lire_tab(corps, chaine, colonne);
            if (test>0) /* on contracte */
            {
                cop_ch(lst_tps, contrali(lst_tps, lst_mots, i));
                cou_dico=lst_mots[i]->motdico;
            }
        }
    }
}

```

```

        cop_ch(date, cou_dico->temps) ;
        if ((!(comp_ch(date, "DJ")) ||
            (!(comp_ch(date, "NJ"))))
            if ((i)&&(!(comp_ch(avant,
"LE")) || !(comp_ch(avant, "CE")) || !(comp_ch(avant, "UN")))))
                cop_ch(lst_tps, contrali(lst_tps,
lst_mots, --i)) ;

        if ((!(comp_ch(date, "DI")) ||
            (!(comp_ch(date, "QA"))))
            if ((i)&&(!(comp_ch(avant, "LE")))))
                cop_ch(lst_tps, contrali(lst_tps,
lst_mots, --i)) ;

        i-- ;
    }
}

if (!(comp_ch(chaine, "CAMO")))
{
    cop_ch(temps, "MO") ;
    cop_ch(entete, entetcor(lst_mots[i + 1])) ;
    colonne = lire_ent(temps, entete) ;
    if ((cou_mots = lst_mots[i] -> precedent))
    {
        if ((cou_dico = cou_mots-> motdico))
        {
            if ((!(comp_ch(cou_dico->temps,
"LE")) || !(comp_ch(cou_dico->temps, "CE")) ||
            !(comp_ch(cou_dico->temps, "UN")) ||
            !(comp_ch(cou_dico->temps, "DN"))))
                cop_ch(corps, cou_dico->temps) ;
            else cop_ch(corps, "") ;
        }
        else
        {
            cou_mots = cou_mots-> precedent ;
            cou_dico = cou_mots-> motdico ;
            if ((!(comp_ch(cou_dico->temps,
"PRLE")) || !(comp_ch(cou_dico->temps, "DELE"))))
                cop_ch(corps, "LE") ;
            else cop_ch(corps, "") ;
        }
    }
    else cop_ch(corps, "") ;
    cou_mots = lst_mots[i] ;
    cou_dico = cou_mots-> motdico ;
    do
    {
        cat_ch(corps, cou_dico->temps) ;
        cou_mots = cou_mots-> decalé ;
        cou_dico = cou_mots-> motdico ;
    }
    while ((cou_mots)&&(cou_dico->mot[0] == '@')) ;
    test = lire_tab(corps, chaine, colonne) ;
    if (test > 0)
    {
        cop_ch(lst_tps, contrali(lst_tps, lst_mots, i + 1)) ;
        i-- ;
    }
}
}

```

```

        if (test <= 0)
        {
            cop_ch(lst_tps, ecralist(lst_tps, lst_mots, i, pre_mots));
            i=0;
        }
    }
}
return(lst_tps);
}

```

/* contracte les combinaisons de déterminants */
 /* de la liste en un seul (le dernier) */

char *contr_dd(lst_tps, lst_mots, pre_mots)

```

char lst_tps[MAXLIST*2+1]; /* chaîne des champs temps */
MOTS *lst_mots[MAXLIST], /* tableau de pointeurs */
    *pre_mots; /* vers les MOTS de la liste */
{
    int longueur, /* de la liste */
        test, /* l'acceptation des formes étudiées */
        colonne, /* de la table */
        i, j; /* indentation */
    char chaine1[5],
        chaine2[7], /* partie de chaîne à comparer */
        temps[MAXTEMPS+1], /* table d'en-tête */
        entete[MAXMOT+1], /* mots à */
        corps[MAXMOT+1]; /* chercher */
    extern char *part_ch(), *contrali(), *ecralist(),
        *entetcor();
    extern lire_tab(), lire_ent();

    longueur=1 ch(lst_tps);
    for (i=0; i<longueur/2; i++)
    {
        test=0;
        cop_ch(chaine1, part_ch(lst_tps, 2*i, 2*i+3));
        cop_ch(chaine2, part_ch(lst_tps, 2*i, 2*i+5));
        if ((!(comp_ch(chaine1, "AVDT")) ||
            (!(comp_ch(chaine1, "AVPD")) ||
            (!(comp_ch(chaine1, "DTDT")) ||
            (!(comp_ch(chaine1, "PDPD")) ||
            (!(comp_ch(chaine1, "DTPD")) ||
            (!(comp_ch(chaine1, "PDDT")) ||
            (!(comp_ch(chaine2, "DTDEDT")))))
            {
                /* chaine1 */

                if ((!(comp_ch(chaine1, "AVDT")) ||
                    (!(comp_ch(chaine1, "AVPD")) ||
                    (!(comp_ch(chaine1, "DTDT")) ||
                    (!(comp_ch(chaine1, "PDPD")) ||
                    (!(comp_ch(chaine1, "DTPD")) ||
                    (!(comp_ch(chaine1, "PDDT")))))
                    {
                        cop_ch(temps, part_ch(lst_tps, 2*i+2, 2*i+3));
                        cop_ch(entete, entetcor(lst_mots[i+1]));
                        colonne=lire_ent(temps, entete);
                    }
                }
            }
        }
    }
}

```

```

        cop_ch(corps, entetcor(lst_mots[i]));
        test=lire_tab(corps, chaîne1, colonne);
        if (test>0) cop_ch(lst_tps, contrali(lst_tps, lst_mots, i--));
    }
    /* chaîne2 */

    if (!(comp_ch(chaîne2, "DTDEDT")))
    {
        cop_ch(temps, part_ch(lst_tps, 2*i+4, 2*i+5));
        cop_ch(entete, entetcor(lst_mots[i+2]));
        colonne=lire_ent(temps, entete);
        cop_ch(corps, entetcor(lst_mots[i]));
        test=lire_tab(corps, chaîne2, colonne);
        if (test>0)
        {
            for (j=0; j<2; j++) cop_ch(lst_tps, contrali(lst_tps,
lst_mots, j));
            i--;
        }
    }

    if (test<=0)
    {
        cop_ch(lst_tps, ecralist(lst_tps, lst_mots, i, pre_mots));
        i=0;
    }
}
return(lst_tps);
}

```

/* contracte la combinaison Prép Dét en Dét */

char *contr_pd(lst_tps, lst_mots, pre_mots)

```

char lst_tps[MAXLIST*2+1]; /* chaîne des champs temps */
MOTS *lst_mots[MAXLIST], /* tableau de pointeurs */
    *pre_mots; /* vers les MOTS de la liste */
{
    int longueur, /* de la liste */
        test, /* l'acceptation des formes étudiées */
        colonne, /* de la table */
        i; /* indentation */
    char chaîne[5], /* partie de chaîne à comparer */
        temps[MAXTEMPS+1], /* table d'en-tête */
        entete[MAXMOT+1], /* mots à */
        corps[MAXMOT+1]; /* chercher */
    extern char *part_ch(), *contrali(), *ecralist(),
        *entetcor();
    extern lire_tab(), lire_ent();

    longueur=1 ch(lst_tps);
    for (i=0; i<longueur/2; i++)
    {
        test=0;
        cop_ch(chaîne, part_ch(lst_tps, 2*i, 2*i+3));
    }
}

```

```

if (((!(comp_ch(chaine, "PRDT")))|
      (!(comp_ch(chaine, "PRPD")))|
      (!(comp_ch(chaine, "PDPR")))|
      (!(comp_ch(chaine, "PRUN")))))
{
    cop_ch(temps, part_ch(lst_tps, 2*i+2, 2*i+3));
    cop_ch(entete, entetcor(lst_mots[i+1]));
    colonne=lire_ent(temps, entete);
    cop_ch(corps, entetcor(lst_mots[i]));
    test=lire_tab(corps, chaine, colonne);
    if (test>0) cop_ch(lst_tps, contrali(lst_tps, lst_mots, i--));
    if (test<0)
    {
        cop_ch(lst_tps, ecralist(lst_tps, lst_mots, i, pre_mots));
        i=0;
    }
}
}
return(lst_tps);
}

```

/* contracte les déterminants de la liste */

char *contr_dt(lst_tps, lst_mots, pre_mots)

```

char lst_tps[MAXLIST*2+1]; /* chaîne des champs temps */
MOTS *lst_mots[MAXLIST], /* tableau de pointeurs */
    *pre_mots; /* vers les MOTS de la liste */
{
    int longueur, /* de la liste */
        test, /* l'acceptation des formes étudiées */
        colonne, /* de la table */
        prep=0, /* existence d'une préposition */
        /* il faut contracter seulement le déterminant */
        i, j; /* indentation */
    char chaine1[5],
        chaine2[7],
        chaine3[9],
        chaine4[11], /* partie de chaîne à comparer */
        temps[MAXTEMPS+1], /* table d'en-tête */
        entete[MAXMOT+1], /* mots à */
        corps[MAXMOT+1]; /* chercher */
    MOTS *cou_mots;
    DICO *cou_dico;
    extern char *part_ch(), *contrali(), *ecralist(),
        *entetcor();
    extern lire_tab(), lire_ent();

    longueur=l ch(lst_tps);
    for (i=0; i<longueur/2; i++)
    {
        test=0;
        cop_ch(chaine1, part_ch(lst_tps, 2*i, 2*i+3));
        cop_ch(chaine2, part_ch(lst_tps, 2*i, 2*i+5));
        cop_ch(chaine3, part_ch(lst_tps, 2*i, 2*i+7));
        cop_ch(chaine4, part_ch(lst_tps, 2*i, 2*i+9));
    }
}

```

```

if ((!(comp_ch(chaine1, "DTNT")))|
    (!(comp_ch(chaine1, "PDNT")))|
    (!(comp_ch(chaine1, "NTPD")))|
    (!(comp_ch(chaine1, "UNNT")))|
    (!(comp_ch(chaine1, "DTCA")))|
    (!(comp_ch(chaine1, "PDCA")))|
    (!(comp_ch(chaine1, "CAPD")))|
    (!(comp_ch(chaine2, "PDLSNT")))|
    (!(comp_ch(chaine2, "PDCSNT")))|
    (!(comp_ch(chaine2, "PDLENT")))|
    (!(comp_ch(chaine2, "PDCENT")))|
    (!(comp_ch(chaine2, "PDDNNT")))|
    (!(comp_ch(chaine2, "DTDENT")))|
    (!(comp_ch(chaine2, "DTDSNT")))|
    (!(comp_ch(chaine2, "PDDSNT")))|
    (!(comp_ch(chaine2, "DTDECA")))|
    (!(comp_ch(chaine2, "DTDENF")))|
    (!(comp_ch(chaine3, "PDLSDNNT")))|
    (!(comp_ch(chaine3, "PDCSDNNT")))|
    (!(comp_ch(chaine3, "DTDELSNT")))|
    (!(comp_ch(chaine3, "DTDECSNT")))|
    (!(comp_ch(chaine3, "DTDELENT")))|
    (!(comp_ch(chaine3, "DTDECENT")))|
    (!(comp_ch(chaine3, "DTDEDNNT")))|
    (!(comp_ch(chaine3, "DTDELENF")))|
    (!(comp_ch(chaine3, "DTDECENF")))|
    (!(comp_ch(chaine4, "DTDELSDNNT")))|
    (!(comp_ch(chaine4, "DTDECSDNNT"))))
{
    if ((i)&&(!(comp_ch("PR", part_ch(lst_tps, 2*i-2, 2*i-1)))) prep=1 ;

        /* chaine1 sans date */

    if ((!(comp_ch(chaine1, "DTNT")))|
        (!(comp_ch(chaine1, "UNNT")))|
        (!(comp_ch(chaine1, "PDNT")))|
        (!(comp_ch(chaine1, "NTPD"))))
    {
        cop_ch(temps, part_ch(lst_tps, 2*i+2, 2*i+3)) ;
        cop_ch(entete, entetcor(lst_mots[i+1])) ;
        colonne=lire_ent(temps, entete) ;
        cop_ch(corps, entetcor(lst_mots[i])) ;
        test=lire_tab(corps, chaine1, colonne) ;
        if (test>0)
        {
            for (j=i ; j<i+2 ; j++)
            {
                cop_ch(chaine1, part_ch(lst_tps, 2*j, 2*j+1)) ;
                if ((!(comp_ch(chaine1,
"DT")))|(!(comp_ch(chaine1, "UN")))|(!(comp_ch(chaine1, "PD")))) cop_ch(lst_tps,
contrali(lst_tps, lst_mots, j)) ;
            }
            i-- ;
        }
    }
}

```

```

/* chaine1 avec date */
if (((!(comp_ch(chaine1, "DTCA")))) || (!(comp_ch(chaine1,
"PDCA")))))
{
    cop_ch(temps, "CA");
    if ((cou_mots=lst_mots[i+1]->precedent))
    {
        cou_dico=cou_mots->motdico;
        if ((!(comp_ch(cou_dico->temps,
"LE")) || (!(comp_ch(cou_dico->temps, "CE")) || (!(comp_ch(cou_dico->temps,
"UN")) || (!(comp_ch(cou_dico->temps, "DN")))))
            cop_ch(entete, cou_dico->temps);
        else cop_ch(entete, "");
    }
    else cop_ch(entete, "");
    cou_mots=lst_mots[i+1];
    cou_dico=cou_mots->motdico;
    do
    {
        cat_ch(entete, cou_dico->temps);
        cou_mots=cou_mots->decale;
        cou_dico=cou_mots->motdico;
    }
    while ((cou_mots)&&(cou_dico->mot[0] == '@'));
    colonne=lire_ent(temps, entete);
    cop_ch(corps, entetcor(lst_mots[i]));
    test=lire_tab(corps, chaine1, colonne);
    if (test>0) cop_ch(lst_tps, contrali(lst_tps, lst_mots, i--));
}

if (!(comp_ch(chaine1, "CAPD")))
{
    cop_ch(temps, "PD");
    cop_ch(entete, entetcor(lst_mots[i+1]));
    colonne=lire_ent(temps, entete);
    if ((cou_mots=lst_mots[i]->precedent))
    {
        if ((cou_dico=cou_mots->motdico))
        {
            if ((!(comp_ch(cou_dico->temps,
"LE")) || (!(comp_ch(cou_dico->temps, "CE")) ||
(!(comp_ch(cou_dico->temps, "UN")) ||
(!(comp_ch(cou_dico->temps, "DN")))))
                cop_ch(corps, cou_dico->temps);
            else cop_ch(corps, "");
        }
        else
        {
            cou_mots=cou_mots->precedent;
            cou_dico=cou_mots->motdico;
            if ((!(comp_ch(cou_dico->temps,
"PRLE")) || (!(comp_ch(cou_dico->temps, "DELE")))))
                cop_ch(corps, "LE");
            else cop_ch(corps, "");
        }
    }
    else cop_ch(corps, "");
    cou_mots=lst_mots[i];
}

```

```

    cou_dico=cou_mots->motdico ;
    do
    {
        cat_ch(corps, cou_dico->temps) ;
        cou_mots=cou_mots->decale ;
        cou_dico=cou_mots->motdico ;
    }
    while ((cou_mots)&&(cou_dico->mot[0]=='@')) ;
    test=lire_tab(corps, chaîne1, colonne) ;
    if (test>0)
    {
        cop_ch(lst_tps, contrali(lst_tps, lst_mots, i+1)) ;
        i-- ;
    }
}

/* chaîne2 sans date */

if (((!(comp_ch(chaîne2, "PDLSNT"))||
      (!(comp_ch(chaîne2, "PDCSNT"))||
      (!(comp_ch(chaîne2, "PDLENT"))||
      (!(comp_ch(chaîne2, "PDCENT"))||
      (!(comp_ch(chaîne2, "PDDNNT"))||
      (!(comp_ch(chaîne2, "DTDENT"))||
      (!(comp_ch(chaîne2, "DTDENF")))))
{
    cop_ch(temps, part_ch(lst_tps, 2*i+4, 2*i+5)) ;
    cop_ch(entete, entetcor(lst_mots[i+2])) ;
    colonne=lire_ent(temps, entete) ;
    cop_ch(corps, entetcor(lst_mots[i])) ;
    test=lire_tab(corps, chaîne2, colonne) ;
    if (test>0)
    {
        if (!prep) for (j=0 ; j<2 ; j++)
            cop_ch(lst_tps, contrali(lst_tps, lst_mots, i)) ;
        else cop_ch(lst_tps, contrali(lst_tps, lst_mots, i)) ;
        i-- ;
    }
}

/* chaîne2 avec date */

if (!(comp_ch(chaîne2, "DTDECA")))
{
    cop_ch(temps, "CA") ;
    if ((cou_mots=lst_mots[i+2]->precedent))
    {
        if ((cou_dico=cou_mots->motdico))
        {
            if (((!comp_ch(cou_dico->temps,
"LE"))||(!comp_ch(cou_dico->temps, "CE"))||
(!comp_ch(cou_dico->temps, "UN"))||
(!comp_ch(cou_dico->temps, "DN")))) cop_ch(entete, cou_dico->temps) ;
            else cop_ch(entete, "") ;
        }
        else
        {
            cou_mots=cou_mots->precedent ;
            cou_dico=cou_mots->motdico ;
        }
    }
}

```

```

cop_ch(entete, "LE");
        if (!comp_ch(cou_dico->temps, "DELE"))
            else cop_ch(entete, "");
    }
    else cop_ch(entete, "");
    cou_mots=lst_mots[i+2];
    cou_dico=cou_mots->motdico;
    do
    {
        cat_ch(entete, cou_dico->temps);
        cou_mots=cou_mots->decale;
        cou_dico=cou_mots->motdico;
    }
    while ((cou_mots)&&(cou_dico->mot[0]=='@'));
    colonne=lire_ent(temps, entete);
    cop_ch(corps, entetcor(lst_mots[i]));
    test=lire_tab(corps, chaine2, colonne);
    if (test>0)
    {
        for (j=0; j<2; j++)
            cop_ch(lst_tps, contrali(lst_tps, lst_mots, i));
        i--;
    }
}

/* chaine3 */

if (((!comp_ch(chaine3, "PDLSDNNT")) ||
(!comp_ch(chaine3, "PDCSDNNT")) ||
(!comp_ch(chaine3, "DTDELSNT")) ||
(!comp_ch(chaine3, "DTDECSNT")) ||
(!comp_ch(chaine3, "DTDELENT")) ||
(!comp_ch(chaine3, "DTDECENT")) ||
(!comp_ch(chaine3, "DTDEDNNT")) ||
(!comp_ch(chaine3, "DTDELENF")) ||
(!comp_ch(chaine3, "DTDECENF"))))
{
    cop_ch(temps, part_ch(lst_tps, 2*i+6, 2*i+7));
    cop_ch(entete, entetcor(lst_mots[i+3]));
    colonne=lire_ent(temps, entete);
    cop_ch(corps, entetcor(lst_mots[i]));
    test=lire_tab(corps, chaine3, colonne);
    if (test>0)
    {
        if (!prep) for (j=0; j<3; j++)
            cop_ch(lst_tps, contrali(lst_tps, lst_mots, i));
        else cop_ch(lst_tps, contrali(lst_tps, lst_mots, i));
        i--;
    }
}

/* chaine4 */

if (((!comp_ch(chaine4, "DTDELSDNNT")) ||
(!comp_ch(chaine4, "DTDECSDNNT"))))
{
    cop_ch(temps, "NT");
    cop_ch(entete, entetcor(lst_mots[i+4]));
    colonne=lire_ent(temps, entete);

```

```

        cop_ch(corps, entetcor(lst_mots[i]));
        cop_ch(chaine3, part_ch(chaine4, 0, 2));
        cat_ch(chaine3, part_ch(chaine4, 4, 4));
        cat_ch(chaine3, part_ch(chaine4, 6, 9));
        test=lire_tab(corps, chaine3, colonne);
        if (test>0)
        {
            if (!prep) for (j=0 ; j<4 ; j++)
                cop_ch(lst_tps, contrali(lst_tps, lst_mots, i));
            else cop_ch(lst_tps, contrali(lst_tps, lst_mots, i));
            i--;
        }
    }
    if (test<=0)
    {
        cop_ch(lst_tps, ecralist(lst_tps, lst_mots, i, pre_mots));
        i=0;
    }
}
return(lst_tps);
}

```

/* contracte les différents adverbes reliés par de */
/* prend la liste à l'envers */

char *contr_de(lst_tps, lst_mots)

char lst_tps[MAXLIST*2+1]; /* chaîne des champs temps */
MOTS *lst_mots[MAXLIST]; /* tableau de pointeurs */
/* vers les MOTS de la liste */

```

{
    int longueur, /* de la liste */
        test, /* l'acceptation des formes étudiées */
        colonne, /* de la table */
        i, j; /* indentation */
    char chaine[7], /* partie de chaîne à comparer */
        temps[MAXTEMPS+1], /* table d'en-tête */
        entete[MAXMOT+1], /* mots à */
        corps[MAXMOT+1]; /* chercher */
    MOTS *cou_mots;
    DICO *cou_dico;
    extern char *part_ch(), *contrali(), *entetcor();
    extern lire_tab(), lire_ent();

    longueur=l_ch(lst_tps);
    for (i=(longueur/2)-1 ; i>=0 ; i--)
    {
        test=0;
        cop_ch(chaine, part_ch(lst_tps, 2*i-4, 2*i+1));
        if ((!(comp_ch(chaine, "NTDEAD")) ||
            (!(comp_ch(chaine, "NTDENT")) ||
            (!(comp_ch(chaine, "CADENT")) ||
            (!(comp_ch(chaine, "NTDECA")) ||
            (!(comp_ch(chaine, "CADECA")) ||
            (!(comp_ch(chaine, "NJDENF")) ||
            (!(comp_ch(chaine, "NTDENF")) ||
            (!(comp_ch(chaine, "NFDEAD")) ||

```

```

        (!comp_ch(chaine, "NTDEDA")) ||
        (!comp_ch(chaine, "NTDEDB")) ||
        (!comp_ch(chaine, "NFDEDA")) ||
        (!comp_ch(chaine, "NFDEDB")) ||
        (!comp_ch(chaine, "NFDENT"))))
    {
        /* chaine sans date */

        if ((!(comp_ch(chaine, "NTDEAD")) ||
            (!comp_ch(chaine, "NTDENT")) ||
            (!comp_ch(chaine, "NJDENF")) ||
            (!comp_ch(chaine, "NTDENF")) ||
            (!comp_ch(chaine, "NFDEAD")) ||
            (!comp_ch(chaine, "NFDENT"))))
        {
            cop_ch(temps, part_ch(lst_tps, 2*i, 2*i+1));
            cop_ch(entete, entetcor(lst_mots[i]));
            colonne=lire_ent(temps, entete);
            cop_ch(corps, entetcor(lst_mots[i-2]));
            test=lire_tab(corps, chaine, colonne);
        }

        /* chaine avec date */

        if (!(comp_ch(chaine, "NTDECA")))
        {
            cop_ch(temps, "CA");
            if ((cou_mots=lst_mots[i]->precedent))
            {
                if ((cou_dico=cou_mots->motdico))
                {
                    if ((!(comp_ch(cou_dico->temps,
"LE")) || !(comp_ch(cou_dico->temps, "CE")) ||
(!comp_ch(cou_dico->temps, "UN")) ||
(!comp_ch(cou_dico->temps, "DN"))))
                        cop_ch(entete, cou_dico->temps);
                    else cop_ch(entete, "");
                }
                else
                {
                    cou_mots=cou_mots->precedent;
                    cou_dico=cou_mots->motdico;
                    if (!(comp_ch(cou_dico->temps, "DELE")))
                        cop_ch(entete, "LE");
                    else cop_ch(entete, "");
                }
            }
            else cop_ch(entete, "");
            cou_mots=lst_mots[i];
            cou_dico=cou_mots->motdico;
            do
            {
                cat_ch(entete, cou_dico->temps);
                cou_mots=cou_mots->decale;
                cou_dico=cou_mots->motdico;
            }
            while ((cou_mots)&&(cou_dico->mot[0]='@'));
            colonne=lire_ent(temps, entete);
            cop_ch(corps, entetcor(lst_mots[i-2]));
            test=lire_tab(corps, chaine, colonne);
        }
    }

```

```

    }

    if (!(comp_ch(chaine, "CADENT")))
    {
        cop_ch(temps, "NT");
        cop_ch(entete, entetcor(lst_mots[i]));
        colonne = lire_ent(temps, entete);
        cou_mots = lst_mots[i-2];
        cou_dico = cou_mots->motdico;
        do
        {
            cop_ch(corps, cou_dico->temps);
            /* (dernier terme de CA) */
            cou_mots = cou_mots->decale;
            cou_dico = cou_mots->motdico;
        }
        while ((cou_mots)&&(cou_dico->mot[0] == '@'));
        test = lire_tab(corps, chaine, colonne);
    }

    if (!(comp_ch(chaine, "CADECA")))
    {
        cop_ch(temps, "CA");
        if ((cou_mots = lst_mots[i]->precedent))
        {
            if ((cou_dico = cou_mots->motdico))
            {
                if (((!comp_ch(cou_dico->temps,
"LE")) || (!comp_ch(cou_dico->temps, "CE"))) ||
(!comp_ch(cou_dico->temps, "UN"))) ||
(!comp_ch(cou_dico->temps, "DN")))) cop_ch(entete, cou_dico->temps);
                else cop_ch(entete, "");
            }
            else
            {
                cou_mots = cou_mots->precedent;
                cou_dico = cou_mots->motdico;
                if (!comp_ch(cou_dico->temps, "DELE"))
                cop_ch(entete, "LE");
                else cop_ch(entete, "");
            }
        }
        else cop_ch(entete, "");
        cou_mots = lst_mots[i];
        cou_dico = cou_mots->motdico;
        do
        {
            cat_ch(entete, cou_dico->temps);
            cou_mots = cou_mots->decale;
            cou_dico = cou_mots->motdico;
        }
        while ((cou_mots)&&(cou_dico->mot[0] == '@'));
        colonne = lire_ent(temps, entete);
        cou_mots = lst_mots[i-2];
        cou_dico = cou_mots->motdico;
        do
        {

```

```

        cop_ch(corps, cou_dico->temps) ;
        /* (dernier terme de CA) */
        cou_mots=cou_mots->decale ;
        cou_dico=cou_mots->motdico ;
    }
    while ((cou_mots)&&(cou_dico->mot[0]='@')) ;
    test=lire_tab(corps, chaîne, colonne) ;
}
if (((!(comp_ch(chaîne, "NTDEDA")))|
    (!(comp_ch(chaîne, "NTDEDB")))|
    (!(comp_ch(chaîne, "NFDEDA")))|
    (!(comp_ch(chaîne, "NFDEDB")))))
{
    colonne=1 ;
    cop_ch(corps, entetcor(lst_mots[i-2])) ;
    test=lire_tab(corps, chaîne, colonne) ;
}
if (test>0)
{
    for (j=0 ; j<2 ; j++) cop_ch(lst_tps, contrali(lst_tps,
lst_mots, i-1)) ;
    i++ ;
}
else
{
    lst_tps[2*i-2]='$' ;
    lst_tps[2*i-1]='$' ;
}
}
if (((chaîne[4]='$')|((chaîne[1]='$'))&&
(chaîne[2]='D')&&(chaîne[3]='E'))
{
    lst_tps[2*i-2]='$' ;
    lst_tps[2*i-1]='$' ;
}
}
longueur=l ch(lst_tps) ;
if ((lst_tps[longueur-2]='D')&&(lst_tps[longueur-1]='E'))
{
    lst_tps[longueur-2]='$' ;
    lst_tps[longueur-1]='$' ;
}
return(lst_tps) ;
}

```

/* contracte les années considérées comme modificateurs */

char *contr_do(lst_tps, lst_mots, pre_mots)

```

char lst_tps[MAXLIST*2+1] ; /* chaîne des champs temps */
MOTS *lst_mots[MAXLIST], /* tableau de pointeurs */
    *pre_mots ; /* vers les MOTS de la liste */
{
    int longueur, /* de la liste */
        test, /* l'acceptation des formes étudiées */
        colonne, /* de la table */
        i ; /* indentation */

```

```

char chaine[5], /* partie de chaîne à comparer */
      corps[MAXMOT+1]; /* mot à chercher */
extern char *part_ch(), *contrali(), *ecralist(),
            *entetcor();
extern lire_tab(), lire_ent();

longueur=l_ch(lst_tps);
for (i=0 ; i<longueur/2 ; i++)
{
    test=0 ;
    cop_ch(chaine, part_ch(lst_tps, 2*i, 2*i+3)) ;
    if ((!(comp_ch(chaine, "NTDA")) ||
          (!(comp_ch(chaine, "NTDB")) ||
            (!(comp_ch(chaine, "NTDO")) ||
              (!(comp_ch(chaine, "NFDA")) ||
                (!(comp_ch(chaine, "NFDB")) ||
                  (!(comp_ch(chaine, "NFDO")))))
            )
          )
        )
    {
        colonne=1 ;
        cop_ch(corps, entetcor(lst_mots[i])) ;
        test=lire_tab(corps, chaine, colonne) ;
        if (test>0)
        {
            cop_ch(lst_tps, contrali(lst_tps, lst_mots, i+1)) ;
            i-- ;
        }
        else
        {
            cop_ch(lst_tps, ecralist(lst_tps, lst_mots, i, pre_mots)) ;
            i=0 ;
        }
    }
}
return(lst_tps) ;
}

```

/* contracte les groupes prépositionnels */

char *contr_gp(lst_tps, lst_mots, pre_mots)

```

char lst_tps[MAXLIST*2+1] ; /* chaîne des champs temps */
MOTS *lst_mots[MAXLIST], /* tableau de pointeurs */
      *pre_mots ; /* vers les MOTS de la liste */
{
    int longueur, /* de la liste */
        test, /* l'acceptation des formes étudiées */
        colonne, /* de la table */
        i, j ; /* indentation */
    char chaine1[5],
          chaine2[7],
          chaine3[9], /* partie de chaîne à comparer */
          temps[MAXTEMPS+1], /* table d'en-tête */
          entete[MAXMOT+1], /* mots à */
          corps[MAXMOT+1] ; /* chercher */
    MOTS *cou_mots ;
    DICO *cou_dico ;
    extern char *part_ch(), *contrali(), *ecralist(),
              *entetcor();

```

```

extern lire_tab(), lire_ent();

longueur=1 ch(lst tps);
for (i=0; i<longueur/2; i++)
{
    test=0;
    cop_ch(chaine1, part_ch(lst tps, 2*i, 2*i+3));
    cop_ch(chaine2, part_ch(lst tps, 2*i, 2*i+5));
    cop_ch(chaine3, part_ch(lst tps, 2*i, 2*i+7));
    if ((!(comp_ch(chaine1, "PRNT")))|
        (!(comp_ch(chaine1, "PRAD")))|
        (!(comp_ch(chaine1, "PRCA")))|
        (!(comp_ch(chaine1, "PRNF")))|
        (!(comp_ch(chaine2, "PRLSNT")))|
        (!(comp_ch(chaine2, "PRCSNT")))|
        (!(comp_ch(chaine2, "PRLENT")))|
        (!(comp_ch(chaine2, "PRCENT")))|
        (!(comp_ch(chaine2, "PRDSNT")))|
        (!(comp_ch(chaine2, "PRDNNT")))|
        (!(comp_ch(chaine2, "PRLNF")))|
        (!(comp_ch(chaine2, "PRCENF")))|
        (!(comp_ch(chaine3, "PRLSDNNT")))|
        (!(comp_ch(chaine3, "PRCSDNNT")))))
    {
        /* chaine1 sans date */

        if ((!(comp_ch(chaine1, "PRNT")))|
            (!(comp_ch(chaine1, "PRAD")))|
            (!(comp_ch(chaine1, "PRNF")))))
        {
            cop_ch(temps, part_ch(lst tps, 2*i+2, 2*i+3));
            cop_ch(entete, entetcor(lst mots[i+1]));
            colonne=lire_ent(temps, entete);
            cop_ch(corps, entetcor(lst mots[i]));
            test=lire_tab(corps, chaine1, colonne);
            if (test>0) cop_ch(lst tps, contrali(lst tps, lst mots, i--));
        }

        /* chaine1 avec date */

        if (!(comp_ch(chaine1, "PRCA")))
        {
            cop_ch(temps, "CA");
            if ((cou_mots=lst mots[i+1]->precedent))
            {
                if ((cou_dico=cou_mots->motdico))
                {
                    if ((!(comp_ch(cou_dico->temps,
"LE")))|
                        (!(comp_ch(cou_dico->temps, "CE")))|
                        (!(comp_ch(cou_dico->temps, "UN")))|
                        (!(comp_ch(cou_dico->temps, "DN"))))
                        cop_ch(entete, cou_dico->temps);
                    else cop_ch(entete, "");
                }
                else
                {
                    cou_mots=cou_mots->precedent;
                    cou_dico=cou_mots->motdico;
                    if (!(comp_ch(cou_dico->temps, "PRLE")))
                    cop_ch(entete, "LE");
                }
            }
        }
    }
}

```

```

        else cop_ch(entete, "");
    }
}
else cop_ch(entete, "");
cou_mots=lst_mots[i+1];
cou_dico=cou_mots->motdico;
do
{
    cat_ch(entete, cou_dico->temps);
    cou_mots=cou_mots->decale;
    cou_dico=cou_mots->motdico;
}
while ((cou_mots)&&(cou_dico->mot[0]='@'));
colonne=lire_ent(temps, entete);
cop_ch(corps, entetcor(lst_mots[i]));
test=lire_tab(corps, chaîne1, colonne);
if (test>0) cop_ch(lst_tps, contrali(lst_tps, lst_mots, i--));
}

/* chaîne2 */

if (((!(comp_ch(chaîne2, "PRLSNT")))||
(!(comp_ch(chaîne2, "PRCSNT")))||
(!(comp_ch(chaîne2, "PRLNT")))||
(!(comp_ch(chaîne2, "PRCENT")))||
(!(comp_ch(chaîne2, "PRDSNT")))||
(!(comp_ch(chaîne2, "PRDNNT")))||
(!(comp_ch(chaîne2, "PRLNF")))||
(!(comp_ch(chaîne2, "PRCENF")))))
{
    cop_ch(temps, part_ch(lst_tps, 2*i+4, 2*i+5));
    cop_ch(entete, entetcor(lst_mots[i+2]));
    colonne=lire_ent(temps, entete);
    cop_ch(corps, entetcor(lst_mots[i]));
    test=lire_tab(corps, chaîne2, colonne);
    if (test>0)
    {
        for (j=0; j<2; j++) cop_ch(lst_tps, contrali(lst_tps,
lst_mots, i));
        i--;
    }
}

/* chaîne3 */

if (((!(comp_ch(chaîne3, "PRLSDNNT")))||
(!(comp_ch(chaîne3, "PRCSDNNT")))))
{
    cop_ch(temps, part_ch(lst_tps, 2*i+6, 2*i+7));
    cop_ch(entete, entetcor(lst_mots[i+3]));
    colonne=lire_ent(temps, entete);
    cop_ch(corps, entetcor(lst_mots[i]));
    test=lire_tab(corps, chaîne3, colonne);
    if (test>0)
    {
        for (j=0; j<3; j++) cop_ch(lst_tps, contrali(lst_tps,
lst_mots, i));
        i--;
    }
}

```

```

        if (test <= 0)
        {
            cop_ch(lst_tps, ecralist(lst_tps, lst_mots, i, pre_mots)) ;
            i=0;
        }
    }
}
return(lst_tps) ;
}

/* contracte les Ntps considérés comme modificateurs */

char *contr_nt(lst_tps, lst_mots, pre_mots)

char lst_tps[MAXLIST*2+1] ; /* chaîne des champs temps */
MOTS *lst_mots[MAXLIST], /* tableau de pointeurs */
    *pre_mots ; /* vers les MOTS de la liste */
{
    int longueur, /* de la liste */
        test, /* l'acceptation des formes étudiées */
        colonne, /* de la table */
        i ; /* indentation */
    char chaine[5], /* partie de chaîne à comparer */
        temps[MAXTEMPS+1], /* table d'en-tête */
        entete[MAXMOT+1], /* mots à */
        corps[MAXMOT+1] ; /* chercher */
    extern char *part_ch(), *contrali(), *ecralist(),
        *entetcor() ;
    extern lire_tab(), lire_ent() ;

    longueur=l_ch(lst_tps) ;
    for (i=0 ; i<longueur/2 ; i++)
    {
        test=0 ;
        cop_ch(chaine, part_ch(lst_tps, 2*i, 2*i+3)) ;
        if ((!(comp_ch(chaine, "ADNT")) ||
            (!(comp_ch(chaine, "NTNT")) ||
            (!(comp_ch(chaine, "NJNT")))))
        {
            cop_ch(temps, part_ch(lst_tps, 2*i+2, 2*i+3)) ;
            cop_ch(entete, entetcor(lst_mots[i+1])) ;
            colonne=lire_ent(temps, entete) ;
            cop_ch(corps, entetcor(lst_mots[i])) ;
            test=lire_tab(corps, chaine, colonne) ;
            if (test>0)
            {
                cop_ch(lst_tps, contrali(lst_tps, lst_mots, i+1)) ;
                i-- ;
            }
            else
            {
                cop_ch(lst_tps, ecralist(lst_tps, lst_mots, i, pre_mots)) ;
                i=0;
            }
        }
    }
}
return(lst_tps) ;
}

```

/* contracte LE et CE */

char *contr_le(lst_tps, lst_mots)

```
char lst_tps[MAXLIST*2+1]; /* chaîne des champs temps */
MOTS *lst_mots[MAXLIST]; /* tableau de pointeurs */
/* vers les MOTS de la liste */
{
    int longueur, /* de la liste */
        i; /* indentation */
    char chaine[5]; /* partie de chaîne à comparer */
    extern char *part_ch(), *contrali();

    longueur=l_ch(lst_tps);
    for (i=0; i<longueur/2; i++)
    {
        cop_ch(chaine, part_ch(lst_tps, 2*i, 2*i+3));
        if ((!(comp_ch(chaine, "LENT")))|
            (!(comp_ch(chaine, "CENT")))|
            (!(comp_ch(chaine, "LENF")))|
            (!(comp_ch(chaine, "CENF"))))
            cop_ch(lst_tps, contrali(lst_tps, lst_mots, i--));
    }
    return(lst_tps);
}
```

/* contracte les ères */

char *contr_er(lst_tps, lst_mots, pre_mots)

```
char lst_tps[MAXLIST*2+1]; /* chaîne des champs temps */
MOTS *lst_mots[MAXLIST], /* tableau de pointeurs */
    *pre_mots; /* vers les MOTS de la liste */
{
    int longueur, /* de la liste */
        test, /* l'acceptation des formes étudiées */
        colonne, /* de la table */
        i; /* indentation */
    char chaine[5], /* partie de chaîne à comparer */
        temps[MAXTEMPS+1], /* table d'en-tête */
        entete[MAXMOT+1], /* mots à */
        corps[MAXMOT+1]; /* chercher */
    extern char *part_ch(), *contrali(), *ecralist(),
        *entetcor();
    extern lire_tab(), lire_ent();

    longueur=l_ch(lst_tps);
    for (i=0; i<longueur/2; i++)
    {
        test=0;
        cop_ch(chaine, part_ch(lst_tps, 2*i, 2*i+3));
        if ((!(comp_ch(chaine, "NTER")))|
            (!(comp_ch(chaine, "DAER")))|
            (!(comp_ch(chaine, "DBER")))|
            (!(comp_ch(chaine, "DOER"))))
        {
            if (!comp_ch(chaine, "NTER"))
            {

```

```

        cop_ch(temps, part_ch(lst_tps, 2*i+2, 2*i+3));
        cop_ch(entete, entetcor(lst_mots[i+1]));
        colonne=lire_ent(temps, entete);
        cop_ch(corps, entetcor(lst_mots[i]));
        test=lire_tab(corps, chaîne, colonne);
    }
    if ((!(comp_ch(chaîne, "DAER")) ||
        (!(comp_ch(chaîne, "DBER")) ||
        (!(comp_ch(chaîne, "DOER")))))
    {
        colonne=1;
        cop_ch(corps, part_ch(lst_tps, 2*i, 2*i+1));
        test=lire_tab(corps, chaîne, colonne);
    }
    if (test>0)
    {
        cop_ch(lst_tps, contrali(lst_tps, lst_mots, i+1));
        i--;
    }
    else
    {
        cop_ch(lst_tps, ecralist(lst_tps, lst_mots, i, pre_mots));
        i=0;
    }
}
}
return(lst_tps);
}
/* contracte Dnum et demi */

```

char *contr_di(lst_tps, lst_mots)

```

char lst_tps[MAXLIST*2+1]; /* chaîne des champs temps */
MOTS *lst_mots[MAXLIST]; /* tableau de pointeurs */
/* vers les MOTS de la liste */
{
    int longueur, /* de la liste */
        i; /* indentation */
    char chaîne[9]; /* partie de chaîne à comparer */
    extern char *part_ch(), *contrali();

    longueur=l ch(lst_tps);
    for (i=0; i<longueur/2; i++)
    {
        cop_ch(chaîne, part_ch(lst_tps, 2*i, 2*i+7));
        if (!comp_ch(chaîne, "DNNTETDI"))
        {
            cop_ch(lst_tps, contrali(lst_tps, lst_mots, i+3));
            cop_ch(lst_tps, contrali(lst_tps, lst_mots, i+2));
            i--;
        }
    }
    return(lst_tps);
}

```

```

/* en-tête ou corps à chercher dans les tables */

char *entetcor(ptr_mots)
MOTS *ptr_mots ;          /* MOTS à chercher */
{
    char mot[MAXMOT+1] ;
    int longueur,          /* de la chaîne */
        i ;               /* indentation */

    cop_ch(mot, (ptr_mots->motdico)->mot) ;
    while ((mot[0] == '@') || (mot[0] == '&'))
    {
        longueur = 1 ;
        for (i = 0 ; i < longueur ; i++) mot[i] = mot[i+1] ;
    }
    return(mot) ;
}

```

9.9. FICHIER BIBLI.H

```

#include <\msc\include\ctype.h>
#include <\msc\include\malloc.h>
#include <\msc\include\math.h>
#include <\msc\include\stdio.h>
#include <\msc\include\string.h>

/* définitions */

#define FICHIER FILE /* structure de fichier */
#define FIN EOF /* fin de fichier */
#define MAXFICH 12 /* taille maximum d'un nom de fichier */
#define RIEN NULL /* pointe vers rien */

/* macros (sauf les tests sur les caractères, voir ci-après) */

#define car_int(x) atoi((x))
#define cat_ch(x,y) strcat((x),(y))
#define comp_ch(x,y) strcmp((x),(y))
#define cop_ch(x,y) strcpy((x),(y))
#define ec_car_e(x) putchar((x))
#define écrit(x,y,z) write((x),(y),(z))
#define ferme(x) close((x))
#define ferme_f(x) fclose((x))
#define ini_f(x) rewind((x))
#define l_ch(x) strlen((x))
#define libere(x) free((x))
#define lit_ca_f(x) fgetc((x))
#define lit_cl(x) gets((x))
#define mémoire(x) malloc((x))
#define ouvre_ec(x) open((x),1)
#define ouv_a_f(x) fopen((x),"a")
#define ouv_ec_f(x) fopen((x),"w")
#define ouv_l_f(x) fopen((x),"r")
#define pos_debf(x,y) fseek((x),(y),0)
#define taille(x) sizeof(x)
#define termine(x) exit((x))

/* macros de tests sur les caractères */

#define est_lt(x) isalpha((x))
#define est_maj(x) isupper((x))
#define est_min(x) islower((x))
#define est_nb(x) isdigit((x))
#define majuscul(x) toupper((x))
#define minuscule(x) tolower((x))

```

9.10. FICHIER *DEFINIR.H*

```

# include "bibli.h"

# define CHOIX    struct premiers_liste /* voir ci-dessous */

# define DICO     struct dictionnaire    /* voir ci-dessous */

# define INDEX    struct mots_index      /* voir ci-dessous */

# define LIGNINDX    512 /* octets par mot de l'index */

# define MAXCHAIN    50          /* taille maximum des chaînes */
                                   /* à transformer */

# define MAXCHIF     5           /* nombre maximum de chiffres */
                                   /* représentant le nombre de caractères du dictionnaire */

# define MAXDNUM     25          /* nombre maximum de mots simples */
                                   /* d'un déterminant numérique */

# define MAXETATS    44          /* nombre d'états de l'automate */

# define MAXGRAM     21          /* longueur maximale des */
                                   /* informations grammaticales */

# define MAXLIST 20              /* longueur maximale des */
                                   /* listes de MOTS */

# define MAXMOT      30          /* longueur maximale d'un mot */

# define MAXPART     5           /* partage maximal des mots */
                                   /* composés du dictionnaire */

# define MAXPONCT    25          /* longueur maximale des */
                                   /* caractères de ponctuation */

# define MAXTEMPS    4           /* longueur maximale des */
                                   /* informations sur les états */

# define MOTS struct suite_de_mots      /* voir ci-dessous */

# define TRANS struct transition        /* voir ci-dessous */

```

```

extern CHOIX                                /* arbre à remplir par la droite */
{
    DICO *reconnu ; /* définition reconnue par l'automate */
    int etat,        /* n° de l'état d'arrivée */
        etage ;      /* par rapport aux MOTS */
    CHOIX *fils,
        *pere,
        *droite,      /* autre choix possible */
        *gauche ;     /* et son retour */
};

extern DICO
{
    char mot[MAXMOT+1], /* dictionnaire */
        gram[MAXGRAM+1],
        temps[MAXTEMPS+1] ;
    DICO *suivant ;
};

extern INDEX
{
    char mot[MAXMOT+1] ;
    int saut ; /* nombre de caractères avant le mot */
    INDEX *suivant ;
};

extern MOTS
{
    char mot[MAXMOT+1],
        car[MAXPONCT+1] ; /* caractères suivant le mot */
    DICO *motdico ; /* suite de mots du dictionnaire */
    MOTS *precedent, *suivant,
        *decale ; /* MOTS suivant après décalage */
};

extern TRANS
{
    char temps[MAXTEMPS+1] ; /* transition */
    int etat ; /* n° de l'état suivant */
                /* négatif si l'état courant est final */
    TRANS *autre ; /* autre transition */
};

```

10. FICHIERS PRINCIPAUX.

10.1. FICHER *ADVDATE.NOM*

ADVDATE.IND/ADVDATE.DIC/ADVDATE.AUT/ADVDATE.RES/

10.2. FICHER *ADVDATE.IND*

à0|au503|avril1037|ces1544|cinquantièmes2049|début2648|deuxième3201|dnumann
ee3744|dnumzéro4320|entières4901|huit5452|juin6083|la6370|le7102|lendemain7632
|les7679|lundi8192|midi8741|millionièmes9282|neuf9775|octantièmes10422|picoseco
ndes10992|premières11534|quatorze12001|quinze12562|rmmois13196|semaine13784|
si14338|soixantièmes14863|treize15359|trois15864|un16163|unième16683|vingts17250
|

10.3. FICHER *ADVDATE.DIC*

à partir de/(Pré#PR)
à peine/(Dét#PD)
à peu de chose près/(Dét#PD)
à peu près/(Dét#PD)
à tout casser/(Dét#PD)
à venir/(A80:Ams:Afs:Amp:Afp#MO)
à/(Pré#PR)
absolument/(Adv#AV)
an/(N1:Nms#NT)
année/(N21:Nfs#NT)
années/(N21:Nfp#NT)
annuellement/(Adv#AD)
ans/(N1:Nmp#NT)
antan/(Adv#AD)
août/(N1:Nms#NM)
approximativement/(Adv#PD)
après-demain/(Adv#AD)
après Jésus-Christ/(Adv#ER)
après-midi/(N80:Nms:Nfs#NT)
après/(Pré#PR)
ascension/(N21:Nfs#NF)
assez/(Adv#DT)
assomption/(N21:Nfs#NF)
au mieux/(Adv#PD)
au moins/(Adv#PD)
au pire/(Adv#PD)
au pis/(Adv#PD)
au plus/(Adv#PD)
au/(Pré#PR)(Pré#PRLE)
aube/(N21:Nfs#NT)
aubes/(N21:Nfp#NT)
aujourd'hui/(Adv#AD)
aurore/(N21:Nfs#NT)
aurores/(N21:Nfp#NT)
aussi/(Adv#AV)
aussitôt/(Pré#PR)

autant/(Adv#DT)
 automne/(N1:Nms#NT)
 automnes/(N1:Nmp#NT)
 aux/(Pré#PR)(Pré#PRLS)
 avant Jésus-Christ/(Adv#ER)
 avant notre ère/(Adv#ER)
 avant tout/(Adv#PD)
 avant-dernier/(A42:Ams#MO)
 avant-derniers/(A42:Amp#MO)
 avant-hier/(Adv#AD)
 avant-veille/(N21:Nfs#NT)
 avant/(Pré#PR)
 avril/(N2S:Nms#NM)
 beaucoup/(Adv#DT)
 bien/(Adv#AV)
 bimensuellement/(Adv#AD)
 brune/(N21:Nfs#NT)
 brunes/(N21:Nfp#NT)
 ce pont/(N1:Nms#DT)
 ce/(Dét:Ams#CE)
 cent/(A80P:Amp:Afp#DN)(A80P:Amp:Afp#DO)(A80P:Amp:Afp#DA)
 centaine/(N21:Nfs#DT)
 centaines/(N21:Nfp#DT)
 centième/(N1:Nms#DT)(A31:Ams:Afs#EM)(A31:Ams:Afs#MO)
 centièmes/(A31:Amp:Afp#EM)(A31:Amp:Afp#MO)(N1:Nmp#DT)
 cents/(A80P:Amp:Afp#DA)(A80P:Amp:Afp#DN)
 certain/(A32:Ams#DT)
 certaine/(A32:Afs#DT)
 certaines/(A32:Afp#DT)
 certains/(A32:Amp#DT)
 ces fêtes/(N21:Nfp#DT)
 ces ponts/(N1:Nmp#DT)
 ces/(Dét:Amp:Afp#CS)
 cet/(Dét:Ams#CE)
 cette fête/(N21:Nfs#DT)
 cette/(Dét:Afs#CE)
 chacun/(N2S:Nms#DT)
 chacune/(N22S:Nfs#DT)
 chaque/(Dét:Ams:Afs#DT)
 ci/(Pro#MO)
 cinq/(A80P:Amp:Afp#DB)(A80P:Amp:Afp#DN)(A80P:Amp:Afp#DJ)(A80P:Amp:Afp#DH)(A80P:Amp:Afp#DK)(A80P:Amp:Afp#DM)
 cinquantaine/(N21:Nfs#DT)
 cinquantaines/(N21:Nfp#DT)
 cinquante/(A80P:Amp:Afp#DN)(A80P:Amp:Afp#DO)(A80P:Amp:Afp#DB)(A80P:Amp:Afp#DK)
 cinquantième/(A31:Ams:Afs#EM)(A31:Ams:Afs#MO)
 cinquantièmes/(A31:Amp:Afp#EM)(A31:Amp:Afp#MO)
 cinquième/(A31:Ams:Afs#EM)(A31:Ams:Afs#MO)
 cinquièmes/(A31:Amp:Afp#EM)(A31:Amp:Afp#MO)
 consécutif/(A38:Ams#MO)
 consécutifs/(A38:Amp#MO)
 consécutive/(A38:Afs#MO)
 consécutives/(A38:Afp#MO)
 courant/(N1:Nms#DT)
 cours/(N2:Nms:Nmp#DT)
 crépuscule/(N1:Nms#NT)
 crépuscules/(N1:Nmp#NT)

d'affilée/(N22S:Nfs#MO)
 d'ici à/(Pré#PR)
 d'ici au/(Pré#PR)(Pré#PRLE)
 d'ici aux/(Pré#PR)(Pré#PRLS)
 d'ici/(Pré#PR)
 d/(Dét#DE)
 dans/(Pré#PR)
 davantage/(Adv#DT)
 de l'hégire/(Adv#ER)
 de notre ère/(Adv#ER)
 de suite/(N21:Nfs#MO)
 de/(Dét#DE)
 début/(N1:Nms#DT)
 débuts/(N1:Nmp#DT)
 décadairement/(Adv#AD)
 décade/(N21:Nfs#NT)
 décades/(N21:Nfp#NT)
 décembre/(N2:Nms:Nmp#NM)
 décennie/(N21:Nfs#NT)
 décennies/(N21:Nfp#NT)
 demain/(Adv#AD)
 demi/(A80:Ams:Afs:Amp:Afp#DI)(A80:Ams:Afs:Amp:Afp#DT)
 demie/(N21:Nfs#DI)
 depuis/(Pré#PR)
 dernier/(A42:Ams#MO)
 dernière/(A42:Afs#MO)
 dernières/(A42:Afp#MO)
 derniers/(A42:Amp#MO)
 des autres/(A31:Amp:Afp#DT)
 des/(Dét:Amp:Afp#DS)(Dét:Amp:Afp#DELS)
 dès/(Pré#PR)
 deux/(A80P:Amp:Afp#DN)(A80P:Amp:Afp#DB)(A80P:Amp:Afp#DJ)(A80P:Amp:Afp#DH)(A80P:Amp:Afp#DK)(A80P:Amp:Afp#DM)
 deuxième demi/(A80:Ams#DT)
 deuxième quart/(N1:Nms#DT)
 deuxième/(A31:Ams:Afs#EM)(A31:Ams:Afs#MO)
 deuxièmes/(A31:Amp:Afp#EM)(A31:Amp:Afp#MO)
 dimanche/(N1:Nms#NJ)(N1:Nms#NT)
 dimanches/(N1:Nmp#NT)
 dix/(A80P:Amp:Afp#DJ)(A80P:Amp:Afp#DN)(A80P:Amp:Afp#DO)(A80P:Amp:Afp#DB)(A80P:Amp:Afp#DM)(A80P:Amp:Afp#DH)(A80P:Amp:Afp#DK)
 dixième/(N1:Nms#DT)(A31:Ams:Afs#EM)(A31:Ams:Afs#MO)
 dixièmes/(N1:Nmp#DT)(A31:Amp:Afp#EM)(A31:Amp:Afp#MO)
 dizaine/(N21:Nfs#DT)
 dizaines/(N21:Nfp#DT)
 dnum/(A80P:Amp:Afp#DN)
 dnuman/(A80P:Amp:Afp#DN)(A80P:Amp:Afp#DB)
 dnumannee/(A80P:Amp:Afp#DN)(A80P:Amp:Afp#DA)
 dnumdecennie/(A80P:Amp:Afp#DN)(A80P:Amp:Afp#DO)(A80P:Amp:Afp#DB)
 dnumheure/(A80P:Amp:Afp#DK)(A80P:Amp:Afp#DJ)(A80P:Amp:Afp#DB)(A80P:Amp:Afp#DH)(A80P:Amp:Afp#DN)
 dnumjour/(A80P:Amp:Afp#DB)(A80P:Amp:Afp#DJ)(A80P:Amp:Afp#DN)(A80P:Amp:Afp#DK)
 dnumminute/(A80P:Amp:Afp#DB)(A80P:Amp:Afp#DN)(A80P:Amp:Afp#DK)
 dnummois/(A80P:Amp:Afp#DM)(A80P:Amp:Afp#DB)(A80P:Amp:Afp#DJ)(A80P:Amp:Afp#DH)(A80P:Amp:Afp#DN)(A80P:Amp:Afp#DK)
 dnumun/(A80P:Amp:Afp#DM)(A80P:Amp:Afp#DB)(A80P:Amp:Afp#DJ)(A80P:Amp:Afp#DH)(A80P:Amp:Afp#DN)(A80P:Amp:Afp#DK)

dnumzéro/(A1:Ams#DN)(A1:Ams#DH)(A80P:Amp:Afp#DK)
 douzaine/(N21:Nfs#DT)
 douzaines/(N21:Nfp#DT)
 douze/(A80P:Amp:Afp#DJ)(A80P:Amp:Afp#DM)(A80P:Amp:Afp#DB)(A80P:Amp:Afp#DN)(A80P:Amp:Afp#DH)(A80P:Amp:Afp#DK)
 douzième/(A31:Ams:Afs#EM)(A31:Ams:Afs#MO)
 douzièmes/(A31:Amp:Afp#EM)(A31:Amp:Afp#MO)
 du/(Dé#DE)(Dé#DELE)
 durant/(Pré#PR)
 écoulé/(V3:Kms#MO)
 écoulée/(V3:Kfs#MO)
 écoulées/(V3:Kfp#MO)
 écoulés/(V3:Kmp#MO)
 ème/(A31:Ams:Afs#MO)
 en gros/(Adv#PD)
 en tout et pour tout/(Adv#PD)
 en tout/(Adv#PD)
 en/(Pré#PR)
 ensuite/(Adv#PD)
 entier/(A42:Ams#MO)
 entière/(A42:Afs#MO)
 entières/(A42:Afp#MO)
 entiers/(A42:Amp#MO)
 entre/(Pré#PR)
 environ/(Adv#PD)
 épiphanie/(N21:Nfs#NF)
 ère/(N21:Nfs#NT)
 ères/(N21:Nfp#NT)
 et/(CCo#ET)(CCo#CJ)
 été/(N1:Nms#NT)
 étés/(N1:Nmp#NT)
 exactement/(Adv#PD)
 férie/(N21:Nfs#NT)
 feries/(N21:Nfp#NT)
 fête/(N21:Nfs#DT)
 fêtes/(N21:Nfp#DT)
 février/(N1:Nms#NM)
 fin/(N21:Nfs#DT)
 fins/(N21:Nfp#DT)
 fort/(Adv#AV)
 gros/(Adv#DT)
 h/(N21:Nfs:Nfp#HE)
 hebdomadairement/(Adv#AD)
 heure/(N21:Nfs#NT)(N21:Nfs#HE)
 heures/(N21:Nfp#NT)(N21:Nfp#HE)
 hier/(Adv#AD)
 hiver/(N1:Nms#NT)
 hivers/(N1:Nmp#NT)
 huit/(A80P:Amp:Afp#DB)(A80P:Amp:Afp#DN)(A80P:Amp:Afp#DJ)(A80P:Amp:Afp#DM)(A80P:Amp:Afp#DH)(A80P:Amp:Afp#DK)
 huitaine/(N21:Nfs#DT)
 huitaines/(N21:Nfp#DT)
 huitante/(A80P:Amp:Afp#DB)(A80P:Amp:Afp#DN)
 huitantième/(A31:Ams:Afs#EM)(A31:Ams:Afs#MO)
 huitantièmes/(A31:Amp:Afp#EM)(A31:Amp:Afp#MO)
 huitième/(A31:Ams:Afs#EM)(A31:Ams:Afs#MO)
 huitièmes/(A31:Amp:Afp#EM)(A31:Amp:Afp#MO)
 janvier/(N1:Nms#NM)

jeudi saint/(N1:Nms#NF)
 jeudi/(N1:Nms#NT)(N1:Nms#NJ)
 jeudis saints/(N1:Nmp#NF)
 jeudis/(N1:Nmp#NT)
 jour/(N1:Nms#NT)
 journée/(N21:Nfs#NT)
 journées/(N21:Nfp#NT)
 journellement/(Adv#AD)
 jours/(N1:Nmp#NT)
 juillet/(N1:Nms#NM)
 juin/(N1:Nms#NM)
 jusqu'à/(Pré#PR)
 jusqu'au/(Pré#PR)(Pré#PRLE)
 jusqu'aux/(Pré#PR)(Pré#PRLS)
 jusqu'en/(Pré#PR)
 jusque vers/(Pré#PR)
 jusque/(Pré#PR)
 l'autre/(A31:Ams:Afs#DT)
 l'extrême fin/(N21:Nfs#DT)
 l'orée/(N21:Nfs#DT)
 l'un/(Dét:Ams#DT)
 l'une/(Dét:Afs#DT)
 l/(Dét:Ams:Afs#LE)
 la centaine/(N21:Nfs#DT)
 la cinquantaine/(N21:Nfs#DT)
 la demi/(A80:Afs#DT)
 la deuxième demi/(A80:Afs#DT)
 la deuxième moitié/(N21:Nfs#DT)
 la dizaine/(N21:Nfs#DT)
 la douzaine/(N21:Nfs#DT)
 la fête/(N21:Nfs#DT)
 la fin/(N21:Nfs#DT)
 la huitaine/(N21:Nfs#DT)
 la mi/(A22S:Afs#DT)
 la moitié/(N21:Nfs#DT)
 la myriade/(N21:Nfs#DT)
 la pleine fin/(N21:Nfs#DT)
 la pleine mi/(A22S:Afs#DT)
 la plupart/(N22S:Nfs#DT)
 la première demi/(A80:Afs#DT)
 la première moitié/(N21:Nfs#DT)
 la quarantaine/(N21:Nfs#DT)
 la quinzaine/(N21:Nfs#DT)
 la seconde demi/(A80:Afs#DT)
 la seconde moitié/(N21:Nfs#DT)
 la seule/(A32:Afs#DT)
 la soixantaine/(N21:Nfs#DT)
 la trentaine/(N21:Nfs#DT)
 la vingtaine/(N21:Nfs#DT)
 là/(Adv#MO)
 la/(Dét:Afs#LE)
 le courant/(N1:Nms#DT)
 le cours/(N2:Nms:Nmp#DT)
 le début/(N1:Nms#DT)
 le demi/(A80:Ams#DT)
 le deuxième demi/(A80:Ams#DT)
 le deuxième quart/(N1:Nms#DT)
 le gros/(Adv#DT)

le milieu/(N3:Nms#DT)
 le peu/(Adv#DT)
 le plein milieu/(N3:Nms#DT)
 le plus clair/(Adv#DT)
 le plus gros/(Adv#DT)
 le pont/(N1:Nms#DT)
 le premier demi/(A80:Ams#DT)
 le premier quart/(N1:Nms#DT)
 le quatrième quart/(N1:Nms#DT)
 le second demi/(A80:Ams#DT)
 le seul/(A32:Ams#DT)
 le tout début/(N1:Nms#DT)
 le troisième quart/(N1:Nms#DT)
 le/(Dét:Ams#LE)
 lendemain/(N1:Nms#NT)
 lendemains/(N1:Nmp#NT)
 les autres/(A31:Amp:Afp#DT)
 les centaines/(N21:Nfp#DT)
 les cinquantaines/(N21:Nfp#DT)
 les débuts/(N1:Nmp#DT)
 les dizaines/(N21:Nfp#DT)
 les douzaines/(N21:Nfp#DT)
 les fêtes/(N21:Nfp#DT)
 les fins/(N21:Nfp#DT)
 les huitaines/(N21:Nfp#DT)
 les myriades/(N21:Nfp#DT)
 les ponts/(N1:Nmp#DT)
 les quarantaines/(N21:Nfp#DT)
 les quinzaines/(N21:Nfp#DT)
 les seules/(A32:Afp#DT)
 les seuls/(A32:Amp#DT)
 les soixantaines/(N21:Nfp#DT)
 les trentaines/(N21:Nfp#DT)
 les vingtaines/(N21:Nfp#DT)
 les/(Dét:Amp:Afp#LS)
 lundi/(N1:Nms#NT)(N1:Nms#NJ)
 lundis/(N1:Nmp#NT)
 mai/(N1:Nms#NM)
 mardi gras/(N1:Nms#NF)
 mardi/(N1:Nms#NT)(N1:Nms#NJ)
 mardis gras/(N1:Nmp#NF)
 mardis/(N1:Nmp#NT)
 mars/(N2:Nms:Nmp#NM)
 matin/(N1:Nms#NT)
 matinée/(N21:Nfs#NT)
 matinées/(N21:Nfp#NT)
 matins/(N1:Nmp#NT)
 même/(Adv#PD)(A31:Ams:Afs#MO)
 mêmes/(A31:Amp:Afp#MO)
 mensuellement/(Adv#AD)
 mercredi des cendres/(N1:Nms#NF)
 mercredi/(N1:Nms#NT)(N1:Nms#NJ)
 mercredis des cendres/(N1:Nmp#NF)
 mercredis/(N1:Nmp#NT)
 mi/(A22S:Afs#DT)
 microseconde/(N21:Nfs#NT)
 microsecondes/(N21:Nfp#NT)
 midi/(N1:Nms#MI)(N1:Nms#NT)

mil/(A80P:Amp:Afp#DA)(A80P:Amp:Afp#DN)
 milieu/(N3:Nms#DT)
 mille/(A80P:Amp:Afp#DO)(A80P:Amp:Afp#DN)(A80P:Amp:Afp#DA)
 millénaire/(N1:Nms#DT)
 millénaires/(N1:Nmp#NT)
 milliard/(N1:Nms#DT)
 milliardième/(N1:Nms#DT)(A31:Ams:Afs#MO)
 milliardières/(N1:Nmp#DT)(A31:Amp:Afp#MO)
 milliards/(N1:Nmp#DT)
 millième/(N1:Nms#DT)(A31:Ams:Afs#EM)(A31:Ams:Afs#MO)
 millièmes/(A31:Amp:Afp#EM)(A31:Amp:Afp#MO)(N1:Nmp#DT)
 millier/(N1:Nms#DT)
 milliers/(N1:Nmp#DT)
 million/(N1:Nms#DT)
 millionième/(N1:Nms#DT)(A31:Ams:Afs#MO)
 millionnières/(A31:Amp:Afp#MO)(N1:Nmp#DT)
 millions/(N1:Nmp#DT)
 milliseconde/(N21:Nfs#NT)
 millisecondes/(N21:Nfp#NT)
 minuit/(N80:Nms#MI)
 minute/(N21:Nfs#NT)(N21:Nfs#MN)
 minutes/(N21:Nfp#NT)(N21:Nfp#MN)
 mn/(N21:Nfs:Nfp#MN)
 moins/(Adv#DT)(Pré#MS)
 mois/(N2:Nms:Nmp#NT)
 moitié/(N21:Nfs#DT)
 myriades/(N21:Nfp#DT)
 n'importe quel/(Dét:Ams#DT)
 n'importe quelle/(Dét:Afs#DT)
 n'importe quelles/(Dét:Afp#DT)
 n'importe quels/(Dét:Amp#DT)
 nanoseconde/(N21:Nfs#NT)
 nanosecondes/(N21:Nfp#NT)
 neuf/(A80P:Amp:Afp#DB)(A80P:Amp:Afp#DN)(A80P:Amp:Afp#DJ)(A80P:Amp:Afp#DM)(A80P:Amp:Afp#DH)(A80P:Amp:Afp#DK)
 neuvième/(A31:Ams:Afs#EM)(A31:Ams:Afs#MO)
 neuvièmes/(A31:Amp:Afp#EM)(A31:Amp:Afp#MO)
 ni/(CCo#CJ)
 Noël/(N1:Nms#NF)
 nonante/(A80P:Amp:Afp#DN)(A80P:Amp:Afp#DB)
 nonantième/(A31:Ams:Afs#EM)(A31:Ams:Afs#MO)
 nonantièmes/(A31:Amp:Afp#EM)(A31:Amp:Afp#MO)
 nouveau/(A73:Ams#MO)
 nouveaux/(A73:Amp#MO)
 nouvelle/(A73:Afs#MO)
 nouvelles/(A73:Afp#MO)
 novembre/(N1:Nms#NM)
 nuit/(N21:Nfs#NT)
 nuitée/(N21:Nfs#NT)
 nuitées/(N21:Nfp#NT)
 nuits/(N21:Nfp#NT)
 octante/(A80P:Amp:Afp#DB)(A80P:Amp:Afp#DN)
 octantième/(A31:Ams:Afs#EM)(A31:Ams:Afs#MO)
 octantièmes/(A31:Amp:Afp#EM)(A31:Amp:Afp#MO)
 octobre/(N1:Nms#NM)
 onze/(A80P:Amp:Afp#DJ)(A80P:Amp:Afp#DM)(A80P:Amp:Afp#DN)(A80P:Amp:Afp#DB)(A80P:Amp:Afp#DH)(A80P:Amp:Afp#DK)
 onzième/(A31:Ams:Afs#EM)(A31:Ams:Afs#MO)

onzièmes/(A31:Amp:Afp#EM)(A31:Amp:Afp#MO)
 orée/(N21:Nfs#DT)
 orées/(N21:Nfp#DT)
 ou/(CCo#CJ)
 pâques/(N22P:Nfp#NF)
 par/(Pré#PR)
 pas mal/(Adv#DT)
 pas tout à fait/(à fait Dét#PD)
 passé/(V3:Kms#MO)(Pré#PR)
 passée/(V3:Kfs#MO)
 passées/(V3:Kfp#MO)
 passés/(V3:Kmp#MO)
 pendant/(Pré#PR)
 pentecôte/(N21:Nfs#NF)
 peu/(Adv#DT)
 picoseconde/(N21:Nfs#NT)
 picosecondes/(N21:Nfp#NT)
 pile/(Adv#MO)
 plein milieu/(N3:Nms#DT)
 plein/(A32:Ams#DT)
 pleine/(A32:Nfs#DT)
 plus clair/(Adv#DT)
 plus gros/(Adv#DT)
 plus ou moins/(Adv#AV)
 plus/(Adv#DT)
 plusieurs/(Dét:Amp:Afp#DT)
 plutôt/(Adv#PD)
 pont/(N1:Nms#DT)
 ponts/(N1:Nmp#DT)
 pour/(Pré#PR)
 pratiquement/(Adv#PD)
 précédent/(N32:Nms#MO)
 précédente/(N32:Nfs#MO)
 précédentes/(N32:Nfp#MO)
 précédents/(N32:Nmp#MO)
 précisément/(Adv#PD)
 premier demi/(A80:Ams#DT)
 premier quart/(N1:Nms#DT)
 premier/(A42:Ams#MO)(A42:Afp#DJ)
 première/(A42:Afs#MO)
 premières/(A42:Afp#MO)
 premiers/(A42:Amp#MO)
 près/(Pré#DT)
 presque/(Adv#PD)
 printemps/(N2:Nms:Nmp#NT)
 prochain/(A32:Ams#MO)
 prochaine/(A32:Afs#MO)
 prochaines/(A32:Afp#MO)
 prochains/(A32:Amp#MO)
 quarantaine/(N21:Nfs#DT)
 quarantaines/(N21:Nfp#DT)
 quarante/(A80P:Amp:Afp#DO)(A80P:Amp:Afp#DB)(A80P:Amp:Afp#DN)(A80P:Amp:Afp#DK)
 quarantième/(A31:Ams:Afs#EM)(A31:Ams:Afs#MO)
 quarantièmes/(A31:Amp:Afp#EM)(A31:Amp:Afp#MO)
 quart/(N1:Nms#QA)
 quarts/(N1:Nmp#DT)

quatorze/(A80P:Amp:Afp#DB)(A80P:Amp:Afp#DN)(A80P:Amp:Afp#DJ)(A80P:Amp:Afp#DH)(A80P:Amp:Afp#DK)
 quatorzième/(A31:Ams:Afs#EM)(A31:Ams:Afs#MO)
 quatorzièmes/(A31:Amp:Afp#EM)(A31:Amp:Afp#MO)
 quatre/(A80P:Amp:Afp#DJ)(A80P:Amp:Afp#DM)(A80P:Amp:Afp#DN)(A80P:Amp:Afp#DB)(A80P:Amp:Afp#DH)(A80P:Amp:Afp#DK)
 quatrième quart/(N1:Nms#DT)
 quatrième/(A31:Ams:Afs#EM)(A31:Ams:Afs#MO)
 quatrième/(A31:Amp:Afp#EM)(A31:Amp:Afp#MO)
 quelques-unes/(A32:Afp#DT)
 quelques-uns/(A32:Amp#DT)
 quelques/(Dét:Amp:Afp#DT)
 quinzaine/(N21:Nfs#NT)
 quinzaines/(N21:Nfp#DT)(N21:Nfp#NT)
 quinze/(A80P:Amp:Afp#DJ)(A80P:Amp:Afp#DN)(A80P:Amp:Afp#DB)(A80P:Amp:Afp#DH)(A80P:Amp:Afp#DK)
 quinzième/(A31:Ams:Afs#EM)(A31:Ams:Afs#MO)
 quinzièmes/(A31:Amp:Afp#EM)(A31:Amp:Afp#MO)
 quotidiennement/(Adv#AD)
 rameaux/(N3:Nmp#NF)
 rm/(A80P:Amp:Afp#DN)
 rman/(A80P:Amp:Afp#DN)(A80P:Amp:Afp#DA)
 rmci/(A80P:Amp:Afp#DN)(A80P:Amp:Afp#DA)(Pro#MO)
 rmdix/(A80P:Amp:Afp#DJ)(A80P:Amp:Afp#DN)(A80P:Amp:Afp#DB)(A80P:Amp:Afp#DM)(A80P:Amp:Afp#DH)(A80P:Amp:Afp#DK)(A80P:Amp:Afp#DO)
 rmi/(A32:Ams#DN)(A32:Ams#DM)(A80P:Amp:Afp#DJ)(A80P:Amp:Afp#DA)
 rmjour/(A80P:Amp:Afp#DN)(A80P:Amp:Afp#DJ)(A80P:Amp:Afp#DA)
 rmmi/(A80P:Amp:Afp#DN)(A80P:Amp:Afp#DA)(A80P:Amp:Afp#DT)
 rmmois/(A80P:Amp:Afp#DN)(A80P:Amp:Afp#DM)(A80P:Amp:Afp#DJ)(A80P:Amp:Afp#DA)
 s/(N21:Nfs:Nfp#SE)
 saison/(N21:Nfs#NT)
 saisons/(N21:Nfp#NT)
 samedi saint/(N1:Nms#NF)
 samedi/(N1:Nms#NT)(N1:Nms#NJ)
 samedis saints/(N1:Nmp#NF)
 samedis/(N1:Nmp#NT)
 second demi/(A80:Ams#DT)
 second/(A32:Ams#MO)
 seconde/(N21:Nfs#NT)(N21:Nfs#SE)(A32:Afs#MO)
 secondes/(N21:Nfp#NT)(N21:Nfp#SE)(A32:Afp#MO)
 seconds/(A32:Amp#MO)
 seize/(A80P:Amp:Afp#DB)(A80P:Amp:Afp#DN)(A80P:Amp:Afp#DJ)(A80P:Amp:Afp#DH)(A80P:Amp:Afp#DK)
 seizième/(A31:Ams:Afs#EM)(A31:Ams:Afs#MO)
 seizièmes/(A31:Amp:Afp#EM)(A31:Amp:Afp#MO)
 semaine/(N21:Nfs#NT)
 semaines/(N21:Nfp#NT)
 semestre/(N1:Nms#NT)
 semestres/(N1:Nmp#NT)
 semestriellement/(Adv#AD)
 sept/(A80P:Amp:Afp#DB)(A80P:Amp:Afp#DN)(A80P:Amp:Afp#DJ)(A80P:Amp:Afp#DM)(A80P:Amp:Afp#DH)(A80P:Amp:Afp#DK)
 septante/(A80P:Amp:Afp#DB)(A80P:Amp:Afp#DN)
 septantième/(A31:Ams:Afs#EM)(A31:Ams:Afs#MO)
 septantièmes/(A31:Amp:Afp#EM)(A31:Amp:Afp#MO)
 septembre/(N1:Nms#NM)
 septième/(A31:Ams:Afs#EM)(A31:Ams:Afs#MO)

septièmes/(A31:Amp:Afp#EM)(A31:Amp:Afp#MO)
 seul/(A32:Ams#DT)
 seulement/(Adv#PD)
 seules/(A32:Afp#DT)
 seuls/(A32:Amp#DT)
 si/(Adv#AV)
 siècle/(N1:Nms#NT)
 siècles/(N1:Nmp#NT)
 simplement/(Adv#PD)
 sitôt/(Pré#PR)
 six/(A80P:Amp:Afp#DJ)(A80P:Amp:Afp#DM)(A80P:Amp:Afp#DN)(A80P:Amp:Afp#DB)(A80P:Amp:Afp#DH)(A80P:Amp:Afp#DK)
 sixième/(A31:Ams:Afs#EM)(A31:Ams:Afs#MO)
 sixièmes/(A31:Amp:Afp#EM)(A31:Amp:Afp#MO)
 soir/(N1:Nms#NT)
 soirée/(N21:Nfs#NT)
 soirées/(N21:Nfp#NT)
 soirs/(N1:Nmp#NT)
 soixantaine/(N21:Nfs#DT)
 soixantaines/(N21:Nfp#DT)
 soixante/(A80P:Amp:Afp#DO)(A80P:Amp:Afp#DB)(A80P:Amp:Afp#DN)
 soixantième/(A31:Ams:Afs#EM)(A31:Ams:Afs#MO)
 soixantièmes/(A31:Amp:Afp#EM)(A31:Amp:Afp#MO)
 sous/(Pré#PR)
 suivant/(A32:Ams#MO)
 suivante/(A32:Afs#MO)
 suivantes/(A32:Afp#MO)
 suivants/(A32:Amp#MO)
 sur/(Pré#PR)
 surlendemain/(N1:Nms#NT)
 surlendemain/(N1:Nmp#NT)
 surtout/(Adv#PD)
 tant/(Adv#DT)
 tantôt/(N1:Nms#NT)(Adv#AD)
 tantôts/(N1:Nmp#NT)
 tierce/(N21:Nfs#NT)
 tierces/(N21:Nfp#NT)
 tous/(A75:Amp#PD)
 toussaint/(N21:Nfs#NF)
 tout au plus/(Adv#PD)
 tout début/(N1:Nms#DT)
 tout/(A75:Ams#PD)
 toute/(A75:Afs#PD)
 toutes/(A75:Afp#PD)
 treize/(A80P:Amp:Afp#DB)(A80P:Amp:Afp#DN)(A80P:Amp:Afp#DJ)(A80P:Amp:Afp#DH)(A80P:Amp:Afp#DK)
 treizième/(A31:Ams:Afs#EM)(A31:Ams:Afs#MO)
 treizièmes/(A31:Amp:Afp#EM)(A31:Amp:Afp#MO)
 trentaine/(N21:Nfs#DT)
 trentaines/(N21:Nfp#DT)
 trente/(A80P:Amp:Afp#DJ)(A80P:Amp:Afp#DN)(A80P:Amp:Afp#DO)(A80P:Amp:Afp#DB)(A80P:Amp:Afp#DK)
 trentième/(A31:Ams:Afs#EM)(A31:Ams:Afs#MO)
 trentièmes/(A31:Amp:Afp#EM)(A31:Amp:Afp#MO)
 très/(Adv#AV)
 trimestre/(N1:Nms#NT)
 trimestres/(N1:Nmp#NT)
 trimestriellement/(Adv#AD)

trois quarts/(N1:Nmp#QT)(N1:Nmp#DT)
 trois/(A80P:Amp:Afp#DB)(A80P:Amp:Afp#DN)(A80P:Amp:Afp#DJ)(A80P:Amp:Afp#DH)(A80P:Amp:Afp#DK)
 troisième quart/(N1:Nms#DT)
 troisième/(A31:Ams:Afs#EM)(A31:Ams:Afs#MO)
 troisièmes/(A31:Amp:Afp#EM)(A31:Amp:Afp#MO)
 trop peu/(Adv#DT)
 trop/(Adv#DT)
 un autre/(A31:Ams:Afs#DT)
 un beau/(Dét:Ams#UN)
 un bon nombre/(N1:Nms#DT)
 un certain/(Dét:Ams#UN)
 un grand nombre/(N1:Nms#DT)
 un nombre/(N1:Nms#DT)
 un quart/(N1:Nms#DT)
 un seul/(A32:Ams#DT)
 un/(Dét:Ams#UN)(A32:Ams#DJ)(A32:Ams#DM)(A32:Ams#DB)(A32:Ams#DN)
 une bonne moitié/(N21:Nfs#DT)
 une bonne partie/(N21:Nfs#DT)
 une certaine/(Dét:Afs#UN)
 une grande moitié/(N21:Nfs#DT)
 une grande partie/(N21:Nfs#DT)
 une moitié/(N21:Nfs#DT)
 une partie/(N21:Nfs#DT)
 une/(Dét#UN)(A32:Afs#DN)(A32:Afs#DH)(A80P:Amp:Afp#DK)
 unième/(A31:Ams:Afs#EM)(A31:Ams:Afs#MO)
 unièmes/(A31:Amp:Afs#EM)(A31:Amp:Afs#MO)
 uniquement/(Adv#PD)
 veille/(N21:Nfs#NT)
 veilles/(N21:Nfp#NT)
 vendredi saint/(N1:Nms#NF)
 vendredi/(N1:Nms#NJ)(N1:Nms#NT)
 vendredis saints/(N1:Nmp#NF)
 vendredis/(N1:Nmp#NT)
 vers/(Pré#PR)
 vigile/(N21:Nfs#NT)
 vigiles/(N21:Nfp#NT)
 vingt/(A80P:Amp:Afp#DO)(A80P:Amp:Afp#DB)(A80P:Amp:Afp#DN)(A80P:Amp:Afp#DJ)(A80P:Amp:Afp#DH)(A80P:Amp:Afp#DK)
 vingtaine/(N21:Nfs#DT)
 vingtaines/(N21:Nfp#DT)
 vingtième/(A31:Ams:Afs#EM)(A31:Ams:Afs#MO)
 vingtièmes/(A31:Amp:Afp#EM)(A31:Amp:Afp#MO)
 vingts/(A80P:Amp:Afp#DB)(A80P:Amp:Afp#DN)
 vraiment/(Adv#AV)
 week-end/(N1:Nms#NT)
 week-ends/(N1:Nmp#NT)
 zéro/(A1:Ams#DN)(A1:Ams#DH)(A80P:Amp:Afp#DK)

10.4. FICHER *ADVDATE.AUT*

0PR1:PD6:AV5:DT7:CE14:LE19:UN21:NF9:NJ13:MO25:DJ17:NM26:NT11:DA12:DB
 12:LS27:CS27:DN30:DS37:AD38:DK34:DH39:MI40:MS41/0AV2:DT3:PD3:CE14:LE1
 9:UN21:NF9:NJ13:MO25:DJ17:NM26:NT11:DA12:DB12:LS27:CS27:DN30:DS37:AD3
 8:DK34:DH39:MI40:MS41/0DT3:PD3/0DT3:PD3:DE4:CE14:LE19:UN21:NF9:NJ13:
 MO25:DJ17:NM26:NT11:DA12:DB12:LS27:CS27:DN30:DS37:AD38:DK34:DH39:MI4
 0:MS41/0DT3:CE14:LE19:UN21:NF9:NJ13:MO25:DJ17:NM26:NT11:DA12:DB12:LS
 27:CS27:DN30/0PD6:DT7/0PR1:PD6:CE14:LE19:UN21:NF9:NJ13:MO25:DJ17:NM2
 6:NT11:DA12:DB12:LS27:CS27:DN30:DS37:AD38:DK34:DH39:MI40:MS41/0DT7:D
 E8:CE14:LE19:UN21:NF9:NJ13:MO25:DJ17:NM26:NT11:DA12:DB12:LS27:CS27:DN
 30:DS37:AD38:DK34:DH39:MI40:MS41/0DT7:CE14:LE19:UN21:NF9:NJ13:MO25:D
 J17:NM26:NT11:DA12:DB12:LS27:CS27:DN30/1MO10:PD10/1PD10/1PD10:ER10:D
 A12:NT9:DB12:DO12:MO10/1ER10:PD10/1MO10:PD10:DJ17:NT16/0NF9:NT11:NJ
 13:MO15:DJ17/0NF9:NT11:NJ13:DJ17/1MO10:PD10:DJ17/1PD10:NM18:DM18:MO
 10/1PD10:DA12:DB12:MO10/0NF9:NJ13:MO15:DJ17:NT11:DI9:QA9/1MO10:PD10:
 DJ24:NT23/0NF9:NJ20:MO22:DJ24:NT11/0NF9:NJ20:DJ24:NT11/1MO10:PD10:DJ2
 4/1PD10:NM9:MO10/0NF9:NJ13:DJ17:NM26:NT11/1DA12:PD10:DB12:MO10/0NF
 9:MO28:NT11:DN30/0NF9:NT11/0DI10/0NF9:MO31:NT32:SE9:MN35:HE33/0NF9:
 NT32/1ET29:PD10:ER10:MO10:NT9:DA12:DB12:DO12/1DK34:PD10:MO10/1PD10
 :MO10:SE9:MN35/1PD10:MO10:DK36/1PD10:MO10:SE9/0NF9:MO28:NT11/1PD10
 :NT10/0HE40/1MS41:DK34:UN43:ET42:MO10:PD10:QT9/0LE43:UN43/0UN43:QA
 9:DI9/0QA9/

11. FICHIERS D'EN-TETE DES TABLES DE TRANSITIONS.

11.1. FICHER *AD.ENT*

aujourd'hui/tantôt/hier + avant-hier/demain + après-
demain/antan/quotidiennement + journallement + hebdomadairement + décadairement
+ mensuellement + bimensuellement + trimestriellement + semestriellement + annuelleme
nt/

11.2. FICHER *CA.ENT*

LENJ/CENJ/UNNJ/NJ/LENJJDJ+LENJJDJNM+LENJJDJNMDA+LENJJDJNMDB+
LENJJDJDM+LENJJDJMDA+LENJJDJMDB/CENJJDJ+CENJJDJNM+CENJJDJN
MDA+CENJJDJNMDB+CENJJDJDM+CENJJDJMDA+CENJJDJMDB/UNNJJDJ
+UNNJJDJNM+UNNJJDJDM/NJDJ+NJDJNM+NJDJNMDA+NJDJNMDB+NJDJ
DM+NJDJMDA+NJDJMDB/LEDJ+LEDJNM+LEDJNMDA+LEDJNMDB+L
EDJDM+LEDJMDA+LEDJMDB/CEDJ+CEDJNM+CEDJNMDA+CEDJNM
DB+CEDJDM+CEDJMDA+CEDJMDB/UNDJ+UNDJNM+UNDJDM/DJ+D
JNM+DJNMDA+DJNMD+DJDM+DJMDA+DJMDB/DK+DKSE+DKMN
+DKMNDK+DKMNDKSE+DHHE+DHEDK+DHEDKMN+DHEDKMND
K+DHEDKMNDKSE+DNSE+DNMN+DNMNDK+DNMNDKSE+DNHE+DN
HEDK+DNHEDKMN+DNHEDKMNDK+DNHEDKMNDKSE+DHHEMSLEQA
+DHHEMSUNQA+DHHEUNQA+DHHEETUNQA+DHHEETQA+DHHEETDI
+DHHEETQ+MSLEQA+MSUNQA/MI+MIDKM+MIDK+MIDKMNDK+MIDK
MNDKSE+MIMSLEQA+MIMSUNQA+MIUNQA+MIETUNQA+MIETQA+MIE
TDI+MIQT/NM+NMDA+NMDB/DA+DADA+DADB/DB+DBDB/



11.3. FICHER DT.ENT

autant/davantage/moins + plus/pas mal/assez/beaucoup/peu/tant/trop + trop
 peu/certain + certaine/chaque/plusieurs/quelques/n'importe quel + n'importe
 quelle + n'importe quels + n'importe quelles/l'autre/le seul + seul + la seule + les
 seuls + les seules/un autre + une autre/un seul + une seule/le demi + la demi + le premier
 demi + la première demi + premier demi + demi/une moitié + une bonne moitié + une
 grande moitié/un nombre + un bon nombre + un grand nombre/une partie + une bonne
 partie + une grande partie/le peu + peu/la plupart/le plus clair + plus clair/le gros + le
 plus gros + gros + plus gros/l'un + l'une/le premier quart + le deuxième quart + le
 troisième quart + le quatrième quart + premier quart + deuxième quart + troisième
 quart + quatrième quart + un quart + trois quarts/la huitaine + les huitaines + la
 dizaine + les dizaines + la douzaine + les douzaines + la quinzaine + les quinzaines + la
 vingtaine + les vingtaines + la trentaine + les trentaines + la quarantaine + les
 quarantaines + la cinquantaine + les cinquantes + la soixantaine + les soixantaines + la
 centaine + les
 centaines + huitaine + dizaine + douzaine + quinzaine + vingtaine + trentaine + quarantaine
 + cinquantaine + soixantaine + centaine + huitaines + dizaines + douzaines + quinzaines + vi
 ngtaines + trentaines + quarantaines + cinquantes + soixantaines + centaines/le
 millier + les milliers + la myriade + les myriades + le million + les millions + le milliard + les
 milliards + milliers + myriades + millions + milliards/chacun + chacune/quelques-
 uns + quelques-unes/près/le cours + cours/le courant + courant/la mi + la pleine
 mi + mi/le début + le plein début + le tout début + début + les débuts + débuts/l'orée/la
 moitié + la première moitié + la seconde moitié + la deuxième moitié/le milieu + le plein
 milieu + milieu/la fin + l'extrême fin + la pleine fin + fin + les fins + fins/plein + pleine/le
 pont + les ponts + ce pont + ces ponts + pont + ponts/la fête + les fêtes + cette fête + ces
 fêtes + fête + fêtes/

11.4. FICHER ER.ENT

avant Jésus-Christ + après Jésus-Christ + avant notre ère + de notre ère + de l'hégire/

11.5. FICHER MO.ENT

prochain + prochaine + prochaines + prochains/précédent + précédentes + précédents + pr
 écédente/passé + passée + passés + passées/écoulé + écoulée + écoulés + écoulées/suivant
 + suivante + suivants + suivantes/dernier + dernière + derniers + dernières/à
 venir/premier + première + premiers + premières/second + seconde/ème + èmes + deuxiè
 me + deuxièmes + troisième + troisièmes + quatrième + quatrièmes + cinquième + cinquièm
 es + sixième + sixièmes + septième + septièmes + huitième + huitièmes + neuvième + neuviè
 mes + dixième + dixièmes + onzième + onzièmes + douzième + douzièmes + treizième + treiz
 ièmes + quatorzième + quatorzièmes + quinzième + quinzièmes + seizième + seizièmes + vin
 gtième + vingtièmes + unième + unièmes + trentième + trentièmes + quarantième + quarant
 ièmes + cinquantième + cinquantièmes + soixantième + soixantièmes + centième + centièm
 es + millième + millièmes + millionième + millionièmes + milliardième + milliardières/ava
 nt-dernier + avant-dernière + avant-derniers + avant-
 dernières/même + mêmes/pile/d'affilée/de
 suite/consécutif + consécutive + consécutives + consécutifs/entier + entière + entiers + enti
 ères/ci/là/nouveau + nouveaux + nouvelle + nouvelles/

11.6. FICHER NF.ENT

épiphany/rameaux/pâques/ascension/pentecôte/assomption/toussaint/noël/

11.7. FICHER NT.ENT

seconde + secondes + tierce + tierces + milliseconde + millisecondes + microseconde + microsecondes + nanoseconde + nanosecondes + picoseconde + picosecondes / minute + minutes / heure + heures / aube + aubes + aurore + aurores / crépuscule + crépuscules + brune + brunes / matin + matins / midi / après-midi / tantôt + tantôts / soir + soirs / nuit + nuits + nuitée + nuitées / matinée + matinées / soirée + soirées / journée + journées / jour + jours + férie + fêtes / dimanche + dimanches + lundi + lundis + mardi + mardis + mercredi + mercredis + jeudi + jeudis + vendredi + vendredis + samedi + samedis / lendemain + lendemains + surlendemain + surlendemain / veille + veilles + vigile + vigiles + avant-veille + avant-veilles / week-end + week-ends / semaine + semaines / décade + décades / quinzaine + quinzaines / mois / trimestre + trimestres + semestre + semestres / saison + saisons / printemps + automne + automnes + hiver + hivers / été + étés / année + années / an + ans / décennie + décennies / siècle + siècles / millénaire + millénaires / ère + ères /

11.8. FICHER PD.ENT

à peine / à peu près + à peu de chose près / au mieux / au moins / au pire / au pis / au plus + tout au plus / à tout casser / en gros / en tout + en tout et pour tout / approximativement / exactement / pratiquement / précisément / seulement / simplement / uniquement / avant tout / pas tout à fait / surtout / tout + toute + tous + toutes / ensuite / environ / même / plutôt / presque /

11.9. FICHER PR.ENT

à + au + aux / à partir de / après / aussitôt + sitôt / avant / dans / depuis / dès d'ici / d'ici à + d'ici au + d'ici aux / durant / en / jusque / jusqu'à + jusqu'au + jusqu'aux / jusqu'en jusque vers / passé / pendant / pour / sous / sur / vers /

11.10. FICHER UN.ENT

un + un certain + un beau /

12.6. FICHIER CAMO.TAB

LENJ:+++++----++--+---/CENJ:+++++----++--+++-
 /UNNJ:+++++----++--+---/NJ:+++++----++-----
 /LENJDJ+LENJDJNM+LENJDJNMDA+LENJDJNMDB+LENJDJDM+LENJDJ
 DMDA+LENJDJDMDB:+++++----+---+---
 /CENJDJ+CENJDJNM+CENJDJNMDA+CENJDJNMDB+CENJDJDM+CENJDJ
 DMDA+CENJDJDMDB:+++++----+---++-
 /UNNJDJ+UNNJDJNM+UNNJDJDM:+++++----+---+---
 /NJDJ+NJDJNM+NJDJNMDA+NJDJNMDB+NJDJDM+NJDJMDA+NJDJDM
 DB:+++++----+-----
 /LEDJ+LEDJNM+LEDJNMDA+LEDJNMDB+LEDJDM+LEDJMDA+LEDJ
 MDB:+++++----+---+---
 /CEDJ+CEDJNM+CEDJNMDA+CEDJNMDB+CEDJDM+CEDJMDA+CEDJ
 DMDB:+++++----+---++-/UNDJ+UNDJNM+UNDJDM:+++++----
 +---+---
 /DJ+DJNM+DJNMDA+DJNMDB+DJDM+DJMDA+DJDMDB:+++++----
 -+-----
 /DK+DKSE+DKMN+DKMNDK+DKMNDKSE+DHHE+DHHEDK+DHHEDK
 MN+DHHEDKMNDK+DHHEDKMNDKSE+DNSE+DNMN+DNMNDK+DNM
 NDKSE+DNHE+DNHEDK+DNHEDKMN+DNHEDKMNDK+DNHEDKMNDK
 SE+DHHEMSLEQA+DHHEMSUNQA+DHHEUNQA+DHHEETUNQA+DHHE
 ETQA+DHHEETDI+DHHEET+MSLEQA+MSUNQA:+++++----
 ++++++--
 /MI+MIDKMN+MIDK+MIDKMNDK+MIDKMNDKSE+MIMSLEQA+MIMSUN
 QA+MIUNQA+MIETUNQA+MIETQA+MIETDI+MIQT/:-++-----++-----
 /NM+NMDA+NMDB:+++++----+-----/DA+DADA+DADB:-++-----
 +-----/DB+DBDB:-++-----+-----/

12.7. FICHER CAPD.TAB

LENJ:++++++
 ++++++/CENJ:++++++
 ++++++/UNNJ:++++++
 ++++++/NJ:++++++
 ++++++/LENJDJ+LENJDJNM+LENJDJNMDA+LENJDJNMDB+LENJDJDM+
 LENJDJDMDA+LENJDJDMDB:++++++
 ++++++/CENJDJ+CENJDJNM+CENJDJNMDA+CENJDJNMDB+CENJDJDM
 +CENJDJDMDA+CENJDJDMDB:++++++
 ++++++/UNNJDJ+UNNJDJNM+UNNJDJDM:++++++
 +-
 ++++++/NJDJ+NJDJNM+NJDJNMDA+NJDJNMDB+NJDJDM+NJDJDMDA+
 NJDJDMDB:++++++
 ++++++/LEDJ+LEDJNM+LEDJNMDA+LEDJNMDB+LEDJDM+LEDJDMDA
 +LEDJDMDB:++++++
 ++++++/CEDJ+CEDJNM+CEDJNMDA+CEDJNMDB+CEDJDM+CEDJDMD
 A+CEDJDMDB:++++++
 ++++++/UNDJ+UNDJNM+UNDJDM:++++++
 ++++++/DJ+DJNM+DJNMDA+DJNMDB+DJDM+DJDMDA+DJDMDB:-----

 /DK+DKSE+DKMN+DKMNDK+DKMNDKSE+DHHE+DHHEKD+DHHEKD
 MN+DHHEKDKNNDK+DHHEKDKNNDKSE+DNSE+DNMN+DNMNDK+DNM
 NDKSE+DNHE+DNHEDK+DNHEDKMN+DNHEDKMNNDK+DNHEDKMNNDK
 SE+DHHEMSLEQA+DHHEMSUNQA+DHHEUNQA+DHHEETUNQA+DHHE
 ETQA+DHHEETDI+DHHEQT+MSLEQA+MSUNQA:++++++
 ++++++-
 ++++++/MI+MIKMN+MIK+MIKMNNDK+MIKMNNDKSE+MIMSLEQA+
 MIMSUNQA+MIUNQA+MIETUNQA+MIETQA+MIETDI+MIQT/
 :++++++
 ++++++/NM+NMDA+NMDB:++++++
 ++++++/DA+DADA+DADB:++++++
 ++++++/DB+DBDB:++++++-+++++/

12.8. FICHER DAER.TAB

DA: +/

12.9. FICHER DTCA.TAB

certain+certaine:---+---+---+--++/chaque:---+---+---+--++/plusieurs:---+---+---
 +--++/quelques:---+---+---+--++/n'importe quel+n'importe quelle+n'importe
 quels+n'importe quelles:---+---+---+--++/l'autre:---+---+---+--++/le
 seul+seul+la seule+les seuls+les seules:---+---+---+--++/un autre+une autre:---+
 --+---+---+--++/le seul+la seule+les seuls+les seules:---+---+---+--++/le
 demi+premier demi+demi+la demi+le premier demi+la première demi:-----
 /le cours+cours:-----/le courant+courant:---+---+---+--++/la mi+la pleine
 mi+mi:---+---+---+--++/le début+le plein début+le tout début+début+les
 débuts+débuts:-----+--++/l'orée:-----+--++/la moitié+la première
 moitié+la seconde moitié+la deuxième moitié:-----+--++/le milieu+le plein
 milieu:-----+--++/la fin+l'extrême fin+la pleine fin+fin+les fins+fins:-----
 +--++/plein+pleine:---+---+---+--++/le pont+les ponts+ce pont+ces
 ponts+pont+ponts:++++++------/la fête+les fêtes+cette fête+ces
 fêtes+fête+fêtes:++++++-

12.10. FICHIER DTDCDNNT.TAB

autant:-----

/davantage:+++++++/moins:++
 ++++++++/plus:++++++
 ++++++++/pas

mal:+++++++/assez:-----

/beaucoup:+++++++/peu:++
 ++++++++/tant:-----

--/trop+trop peu:+++++++/une
 moitié+une bonne moitié+une grande

moitié:+++++++/un

nombre+un bon nombre+un grand

nombre:+++++++/une

partie+une bonne partie+une grande

partie:+++++++/le

peu:+++++++/la

plupart:+++++++/le plus

clair:+++++++/le gros+le plus

gros:+++++++/l'un+l'une:++

+++++++/le premier quart+le

deuxième quart+le troisième quart+le quatrième quart:-----/la

huitaine+les huitaines+la dizaine+les dizaines+la douzaine+les douzaines+la

quinzaine+les quinzaines+la vingtaine+les vingtaines+la trentaine+les trentaines+la

quarantaine+les quarantaines+la cinquantaine+les cinquantaines+la soixantaine+les

soixantaines+la centaine+les centaines:-----/le millier+les

milliers+la myriade+les myriades+le million+les millions+le milliard+les milliards:---

/chacun+chacune:+++++++/qu
 elques-uns+quelques-

unes:+++++++/près:++++
 ++++++++/le

cours+cours:+++++++/le

courant+courant:+++++++/le

début+le plein début+le tout début+début+les

débuts+débuts:+++++++/l'orée:
 -----/la moitié+la première moitié+la seconde moitié+la

deuxième moitié:+++++++/le

milieu+le plein

milieu+milieu:+++++++/la

fin+l'extrême fin+la pleine fin+fin+les

fins+fins:+++++++/plein+plein

e:+++++++/

le pont + les ponts + ce pont + ces ponts + pont + ponts: -- + + + + + / la fête + les fêtes + cette fête + ces fêtes + fête + fêtes: + + - + + + + /

12.13. FICHER DTDECENT.TAB

autant:-----
 /davantage:+++++++ /moins: +
 ++++++ /plus:+++++
 ++++++ /pas mal:-----
 /assez:-----/beaucoup:-----/peu:-----
 -----/tant:-----/trop + trop peu:-----/une
 moitié + une bonne moitié + une grande moitié:++++-
 ++++++ /un nombre + un bon nombre + un grand nombre:----
 -----/une partie + une bonne partie + une grande
 partie:+++++ /le peu + peu:---
 -----/la plupart:-----/le plus clair + plus
 clair:++++-+----- /le gros + gros + plus
 gros + le plus gros:++++-+----- /l'un + l'une:--
 -----/le premier quart + le deuxième quart + le troisième quart + le
 quatrième quart + premier quart + deuxième quart + troisième quart + quatrième
 quart + un quart + trois quarts:++++-+-----
 ++++++ /la huitaine + les huitaines + la dizaine + les
 dizaines + la douzaine + les douzaines + la quinzaine + les quinzaines + la vingtaine + les
 vingtaines + la trentaine + les trentaines + la quarantaine + les quarantaines + la
 cinquantaine + les cinquantaines + la soixantaine + les soixantaines + la centaine + les
 centaines + huitaine + dizaine + douzaine + quinzaine + vingtaine + trentaine + quarantaine
 + cinquantaine + soixantaine + centaine + huitaines + dizaines + douzaines + quinzaines + vi
 ngtaines + trentaines + quarantaines + cinquantaines + soixantaines + centaines:-----
 -----/le millier + les milliers + la myriade + les myriades + le million + les
 millions + le milliard + les milliards + milliers + myriades + millions + milliards:-----
 -----/chacun + chacune:-----/quelques-uns + quelques-unes:-----

 /près:+++++ /le
 cours + cours:+++++ /le
 courant + courant:++++-+----- /le
 début + le plein début + le tout début + début + les débuts + débuts:++++-+-----
 ++++++ /l'orée:++++-+-----
 ++++++ /la moitié + la première moitié + la seconde
 moitié + la deuxième moitié:++++-+-----
 ++++++ /le milieu + le plein milieu + milieu:++++-+-----
 ----- /la fin + l'extrême fin + la pleine
 fin + fin + les fins + fins:++++-+-----
 ++++++ /plein + pleine:-----/

12.14. FICHIER *DTDECSNT.TAB*

autant:-----
 /davantage:+++++++/moins:++
 ++++++++/plus:++++++
 ++++++++/pas
 mal:+++++++/assez:-----

 /beaucoup:+++++++/peu:+++
 ++++++++/tant:-----
 --/trop+ trop peu:+++++++/une
 moitié+ une bonne moitié+ une grande
 moitié:+++++++/un
 nombre+ un bon nombre+ un grand
 nombre:+++++++/une
 partie+ une bonne partie+ une grande
 partie:+++++++/le
 peu+ peu:+++++++/la
 plupart:+++++++/le plus
 clair+ plus clair:+++++++/le
 gros+ gros+ plus gros+ le plus
 gros:+++++++/l'un+l'une:+++
 ++++++++/le premier quart+ le
 deuxième quart+ le troisième quart+ le quatrième quart+ premier quart+ deuxième
 quart+ troisième quart+ quatrième quart+ un quart+ trois
 quarts:+++++++/la huitaine+ les
 huitaines+ la dizaine+ les dizaines+ la douzaine+ les douzaines+ la quinzaine+ les
 quinzaines+ la vingtaine+ les vingtaines+ la trentaine+ les trentaines+ la
 quarantaine+ les quarantaines+ la cinquantaine+ les cinquantes+ la soixantaine+ les
 soixantaines+ la centaine+ les
 centaines+ huitaine+ dizaine+ douzaine+ quinzaine+ vingtaine+ trentaine+ quarantaine
 + cinquantaine+ soixantaine+ centaine+ huitaines+ dizaines+ douzaines+ quinzaines+ vi
 ngtaines+ trentaines+ quarantaines+ cinquantes+ soixantaines+ centaines:-----
 -----/le millier+ les milliers+ la myriade+ les myriades+ le million+ les
 millions+ le milliard+ les milliards+ milliers+ myriades+ millions+ milliards:-----

 /chacun+ chacune:+++++++/qu
 elques-uns+ quelques-
 unes:+++++++/près:+++++
 ++++++++/le
 cours+ cours:+++++++/le
 courant+ courant:+++++++/le
 début+ le plein début+ le tout début+ début+ les
 débuts+ débuts:+++++++/l'orée:
 -----/la moitié+ la première moitié+ la seconde moitié+ la
 deuxième moitié:+++++++/le
 milieu+ le plein
 milieu+ milieu:+++++++/la
 fin+ l'extrême fin+ la pleine fin+ fin+ les
 fins+ fins:+++++++/plein+ plein
 e:+++++++/

12.15. FICHER DTDEDNNT.TAB

autant:-----
 /davantage:+++++++ /moins:++
 ++++++ /plus:+++++
 ++++++ /pas mal:-----
 /assez:+++++++ /beaucoup:-----
 ----- /peu:----- /tant:-----
 /trop + trop peu:----- /une moitié + une bonne moitié + une grande
 moitié:----- /un nombre + un bon nombre + un grand nombre:-----
 ----- /une partie + une bonne partie + une grande partie:-----
 ----- /le peu + peu:----- /la plupart:----- /le plus
 clair + plus clair:----- /le gros + gros + plus gros + le plus gros:-----
 ----- /l'un + l'une:----- /le premier quart + le deuxième
 quart + le troisième quart + le quatrième quart + premier quart + deuxième
 quart + troisième quart + quatrième quart + un quart + trois quarts:-----
 -- /la huitaine + les huitaines + la dizaine + les dizaines + la douzaine + les douzaines + la
 quinzaine + les quinzaines + la vingtaine + les vingtaines + la trentaine + les trentaines + la
 quarantaine + les quarantaines + la cinquantaine + les cinquantaines + la soixantaine + les
 soixantaines + la centaine + les
 centaines + huitaine + dizaine + douzaine + quinzaine + vingtaine + trentaine + quarantaine
 + cinquantaine + soixantaine + centaine + huitaines + dizaines + douzaines + quinzaines + vi
 ngtaines + trentaines + quarantaines + cinquantaines + soixantaines + centaines:-----
 ----- /le millier + les milliers + la myriade + les myriades + le million + les
 millions + le milliard + les milliards + milliers + myriades + millions + milliards:-----
 ----- /chacun + chacune:----- /quelques-uns + quelques-unes:-

 /près:+++++++ /le
 cours + cours:+++++++ /le
 courant + courant:+++++++ /le
 début + le plein début + le tout début + début + les
 débuts + débuts:+++++++ /l'orée:
 ----- /la moitié + la première moitié + la seconde moitié + la
 deuxième moitié:+++-----+-----+----- /le milieu + le plein
 milieu + milieu:+++-----+-----+----- /la fin + l'extrême fin + la
 pleine fin + fin + les
 fins + fins:+++++++ /plein + plein
 e:-----/

[illegible]

le pont + les ponts + ce pont + ces ponts + pont + ponts: --- + + + - / la fête + les fêtes + cette fête + ces fêtes + fête + fêtes: + + - + + + - /

12.18. FICHIER DTDELENT.TAB

autant:-----
 /davantage:+++++ /moins:++
 +++++ /plus:+++++
 +++++ /pas mal:-----
 /assez:-----/beaucoup:-----/peu:-----
 -----/tant:-----/trop + trop peu:-----/une
 moitié + une bonne moitié + une grande moitié:++ +-----++ + -
 +++++ /un nombre + un bon nombre + un grand nombre:-----
 -----/une partie + une bonne partie + une grande
 partie:+++++ /le peu + peu:-----
 -----/la plupart:-----/le plus clair + plus
 clair:++ +-----+---+ +++++ /le gros + gros + plus
 gros + le plus gros:++ +-----+---+ +++++ /l'un + l'une:--
 -----/le premier quart + le deuxième quart + le troisième quart + le
 quatrième quart + premier quart + deuxième quart + troisième quart + quatrième
 quart + un quart + trois quarts:++ +-----+---+ + + -
 +++++ /la huitaine + les huitaines + la dizaine + les
 dizaines + la douzaine + les douzaines + la quinzaine + les quinzaines + la vingtaine + les
 vingtaines + la trentaine + les trentaines + la quarantaine + les quarantaines + la
 cinquantaine + les cinquantaines + la soixantaine + les soixantaines + la centaine + les
 centaines + huitaine + dizaine + douzaine + quinzaine + vingtaine + trentaine + quarantaine
 + cinquantaine + soixantaine + centaine + huitaines + dizaines + douzaines + quinzaines + vi
 ngtaines + trentaines + quarantaines + cinquantaines + soixantaines + centaines:-----
 -----/le millier + les milliers + la myriade + les myriades + le million + les
 millions + le milliard + les milliards + milliers + myriades + millions + milliards:-----
 -----/chacun + chacune:-----/quelques-uns + quelques-unes:-----

 /près:+++++ /le
 cours + cours:+++++ /le
 courant + courant:++ +-----+---+ +++++ /le
 début + le plein début + le tout début + début + les débuts + débuts:++ + + +-----+--
 +++++ /l'orée:++ + + +-----+--
 +++++ /la moitié + la première moitié + la seconde
 moitié + la deuxième moitié:++ +-----+--
 +++++ /le milieu + le plein milieu + milieu:++ + --
 --+---+ +++++ /la fin + l'extrême fin + la pleine
 fin + fin + les fins + fins:++ + + +-----+--
 +++++ /plein + pleine:-----/

12.19. FICHER DTDELSNT.TAB

autant:-----/davantage:-----
 /moins:+++++++/plus:+++++
 ++++++/pas
 mal:+++++++/assez:-----

 /beaucoup:+++++++/peu:+++
 ++++++/tant:-----
 --/trop+trop peu:+++++++/une
 moitié+une bonne moitié+une grande
 moitié:+++++++/un
 nombre+un bon nombre+un grand
 nombre:+++++++/une
 partie+une bonne partie+une grande
 partie:+++++++/le
 peu+peu:+++++++/la
 plupart:+++++++/le plus
 clair+plus clair:+++++++/le
 gros+gros+plus gros+le plus
 gros:+++++++/l'un+l'une:+++
 ++++++/le premier quart+le
 deuxième quart+le troisième quart+le quatrième quart+premier quart+deuxième
 quart+troisième quart+quatrième quart+un quart+trois
 quarts:+++++++/la huitaine+les
 huitaines+la dizaine+les dizaines+la douzaine+les douzaines+la quinzaine+les
 quinzaines+la vingtaine+les vingtaines+la trentaine+les trentaines+la
 quarantaine+les quarantaines+la cinquantaine+les cinquantaines+la soixantaine+les
 soixantaines+la centaine+les
 centaines+huitaine+dizaine+douzaine+quinzaine+vingtaine+trentaine+quarantaine
 +cinquantaine+soixantaine+centaine+huitaines+dizaines+douzaines+quinzaines+vi
 ngtaines+trentaines+quarantaines+cinquantaines+soixantaines+centaines:-----
 -----/le millier+les milliers+la myriade+les myriades+le million+les
 millions+le milliard+les milliards+milliers+myriades+millions+milliards:-----

 /chacun+chacune:+++++++/qu
 elques-uns+quelques-
 unes:+++++++/près:+++++
 ++++++/le
 cours+cours:+++++++/le
 courant+courant:-----/le début+le plein début+le tout
 début+début+les
 débuts+débuts:+++++++/l'orée:
 ++++++/la moitié+la première
 moitié+la seconde moitié+la deuxième
 moitié:+++++++/le milieu+le
 plein
 milieu+milieu:+++++++/la
 fin+l'extrême fin+la pleine fin+fin+les
 fins+fins:+++++++/plein+plein
 e:-----/

12.20. FICHER DTDENF.TAB

le pont+les ponts+ce pont+ces ponts+pont+ponts:--+-+--+ /la fête+les fêtes+cette
 fête+ces fêtes+fête+fêtes:--+-+--+ /

[illegible]

12.22. FICHER DTDLDNNT.TAB

autant:-----
 /davantage:+++++++++/moins:++
 ++++++/plus:++++++
 ++++++/pas
 mal:+++++++/assez:-----

 /beaucoup:+++++++/peu:+++
 ++++++/tant:-----
 --/trop+trop peu:+++++++/une
 moitié+une bonne moitié+une grande
 moitié:+++++++/un
 nombre+un bon nombre+un grand
 nombre:+++++++/une
 partie+une bonne partie+une grande
 partie:+++++++/le
 peu:+++++++/la
 plupart:+++++++/le plus
 clair:+++++++/le gros+le plus
 gros:+++++++/l'un+l'une:++
 ++++++/le premier quart+le
 deuxième quart+le troisième quart+le quatrième quart:-----/la
 huitaine+les huitaines+la dizaine+les dizaines+la douzaine+les douzaines+la
 quinzaine+les quinzaines+la vingtaine+les vingtaines+la trentaine+les trentaines+la
 quarantaine+les quarantaines+la cinquantaine+les cinquantaines+la soixantaine+les
 soixantaines+la centaine+les centaines:-----/le millier+les
 milliers+la myriade+les myriades+le million+les millions+le milliard+les milliards:---

 /chacun+chacune:+++++++/qu
 elques-uns+quelques-
 unes:+++++++/près:++++
 ++++++/le
 cours+cours:+++++++/le
 courant+courant:+++++++/le
 début+le plein début+le tout début+début+les
 débuts+débuts:+++++++/l'orée:
 -----/la moitié+la première moitié+la seconde moitié+la
 deuxième moitié:+++++++/le
 milieu+le plein
 milieu+milieu:+++++++/la
 fin+l'extrême fin+la pleine fin+fin+les
 fins+fins:+++++++/plein+plein
 e:+++++++/

12.23. FICHER DTDSNT.TAB

autant:-----
 /davantage:+++++++/moins:++
 ++++++++/plus:++++++
 ++++++++/pas
 mal:+++++++/assez:-----

 /beaucoup:+++++++/peu:++
 ++++++++/tant:-----
 --/trop + trop peu:+++++++/une
 moitié + une bonne moitié + une grande
 moitié:+++++++/un
 nombre + un bon nombre + un grand
 nombre:+++++++/une
 partie + une bonne partie + une grande
 partie:+++++++/le
 peu:+++++++/la
 plupart:+++++++/le plus
 clair:+++++++/le gros + le plus
 gros:+++++++/l'un + l'une:++
 ++++++++/le premier quart + le
 deuxième quart + le troisième quart + le quatrième quart:-----/la
 huitaine + les huitaines + la dizaine + les dizaines + la douzaine + les douzaines + la
 quinzaine + les quinzaines + la vingtaine + les vingtaines + la trentaine + les trentaines + la
 quarantaine + les quarantaines + la cinquantaine + les cinquantaines + la soixantaine + les
 soixantaines + la centaine + les centaines:-----/le millier + les
 milliers + la myriade + les myriades + le million + les millions + le milliard + les milliards:---

 /chacun + chacune:+++++++/qu
 elques-uns + quelques-
 unes:+++++++/près:++++
 ++++++++/le
 cours + cours:+++++++/le
 courant + courant:+++++++/le
 début + le plein début + le tout début + début + les
 débuts + débuts:+++++++/l'orée:
 -----/la moitié + la première moitié + la seconde moitié + la
 deuxième moitié:+++++++/le
 milieu + le plein
 milieu + milieu:+++++++/la
 fin + l'extrême fin + la pleine fin + fin + les
 fins + fins:+++++++/plein + plein
 e:+++++++/

12.24. FICHER DTDT.TAB

autant:----- + /davantage:-----
 + /moins:----- + /plus:-----
 + /pas mal:----- + /assez:----- +
 - + /beaucoup:-- + ----- + /peu:-----
 ---- + /tant:----- + /trop + trop peu:-----
 ----- + /

prochain + prochains + prochaines: + + + + + + + +
+ + + + + + + + /précédent + précédente + précédents + précédente: + + + + + + + +
+ + + + + + + + /passé + passée + passés + passées:-----
/écoulé + écoulée + écoulés + écoulées:-----
/suivant + suivante + suivants + suivantes:-----
/dernier + dernière + derniers + dernières: + + + + + + + + + + /à venir:-----
-----/premier + première + premiers + premières: + + + + + + + +
+ + + + + + + + /second + seconde: - + + + + + + +
+ + + + + + + + /second + seconde /ème + èmes + deuxième + deuxièmes + troisième + trois
ièmes + quatrième + quatrièmes + cinquième + cinquièmes + sixième + sixièmes + septième
+ septièmes + huitième + huitièmes + neuvième + neuvièmes + dixième + dixièmes + onzièm
e + onzièmes + douzième + douzièmes + treizième + treizièmes + quatorzième + quatorzièm
es + quinzisième + quinzisièmes + seizième + seizièmes + vingtième + vingtièmes + unième + un
ièmes + trentième + trentièmes + quarantième + quarantièmes + cinquantième + cinquantiè
mes + soixantième + soixantièmes + centième + centièmes + millièm + millièmes + millioni
ème + millionièmes + milliardième + milliardières: + + + + + + + +
+ + + + + + + + /avant-dernier + avant-dernière + avant-derniers + avant-
dernières: + + + + + + + +
+ + + + + + + + /même + mêmes: + + + + + + + + + + + + + + /pile:-----
/d'affilée:-----/de suite:-----
/consécutif + consécutive + consécutives + consécutifs:-----
/entier + entière + entiers + entières:----- + + + ---
/nouveau + nouveaux + nouvelle + nouvelles:----- + + + + + + /

12.31. FICHER NFDB.TAB

épiphanie: + /rameaux: + /pâques: + /ascension: + /pentecôte: + /assomption: + /toussaint
: + /noël: + /

12.32. FICHER NFDEAD.TAB

épiphanie: + + + + + - /rameaux: + + + + + - /pâques: + + + + + - /ascension: + + + + + -
/pentecôte: + + + + + - /assomption: + + + + + - /toussaint: + + + + + - /noël: + + + + + - /

12.33. FICHER NFDEDA.TAB

épiphanie: + /rameaux: + /pâques: + /ascension: + /pentecôte: + /assomption: + /toussaint
: + /noël: + /

12.34. FICHER NFDEDB.TAB

épiphanie: + /rameaux: + /pâques: + /ascension: + /pentecôte: + /assomption: + /toussaint
: + /noël: + /

12.35. FICHER NFDENT.TAB

épiphanie: ----- + + + ----- + + ---- /rameaux: ----- + + + ----- + + ----
/pâques: ----- + + + ----- + + ---- /ascension: ----- + + + ----- + + ----
/pentecôte: ----- + + + ----- + + ---- /assomption: ----- + + + ----- + + ----
/toussaint: ----- + + + ----- + + ---- /noël: ----- + + + ----- + + ---- /

12.36. FICHER NFDO.TAB

épiphanie: + /rameaux: + /pâques: + /ascension: + /pentecôte: + /assomption: + /toussaint
: + /noël: + /

12.37. FICHER NFMO.TAB

1er janvier: + + + + + + + + + + + + + + + + - - - /Epiphanie: + + + + + + + + + + + + + + + + - - -
+ + + + + + + + + + + + + + + + - - - /Rameaux: + + + + + + + + + + + + + + + + - - -
/Pâques: + + + + + + + + + + + + + + + + - - - /1er mai: + + + + + + + + + + + + + + + + - - -
+ + + + + + + + + + + + + + + + - - - /8 mai: + + + + + + + + + + + + + + + + - - -
/Ascension: + + + + + + + + + + + + + + + + - - - /Pentecôte: + + + + + + + + + + + + + + + + - - -
+ + + + + + + + + + + + + + + + - - - /14 juillet: + + + + + + + + + + + + + + + + - - -
/Assomption: + + + + + + + + + + + + + + + + - - - /Toussaint: + + + + + + + + + + + + + + + + - - -
+ + + + + + + + + + + + + + + + - - - /1er novembre: + + + + + + + + + + + + + + + + - - - /11
novembre: + + + + + + + + + + + + + + + + - - - /Noël: + + + + + + + + + + + + + + + + - - -
+ + + + + + + + + + + + + + + + - - - /25 décembre: + + + + + + + + + + + + + + + + - - - /

12.38. FICHER NJDENF.TAB

dimanche: - + + + + - + + + + + + + + + + + + /lundi: - + - + + - + + + + + + + + + + + + /mardi: - + - + + -
+ + + + + + + + + + + + + + + + + + /mercredi: - + - + + - + + + + + + + + + + + + /jeudi: - + - + + -
+ + + + + + + + + + + + + + + + + + /vendredi: - + - + + - + + + + + + + + + + + + /samedi: - + - + + - + + + + + + + + + + + + /

12.39. FICHER NJNT.TAB

dimanche + lundi + mardi + mercredi + jeudi + vendredi + samedi: ----- + + + + + ----- + + + -
----- /

12.40. FICHER NTDA.TAB

seconde + secondes + tierce + tierces + milliseconde + millisecondes + microseconde + microsecondes + nanoseconde + nanosecondes + picoseconde + picosecondes:-
 /minute + minutes:-/heure + heures:-/aube + aubes + aurore + aurores:-
 /crépuscule + crépuscules + brune + brunes:-/matin + matins:-/midi:-/après-midi:-
 /tantôt + tantôts:-/soir + soirs:-/nuit + nuits + nuitée + nuitées:-/matinée + matinées:-
 /soirée + soirées:-/journée + journées:-/jour + jours + férie + fêtes:-
 /dimanche + dimanches + lundi + lundis + mardi + mardis + mercredi + mercredis + jeudi + jeudis + vendredi + vendredis + samedi + samedis:-
 /lendemain + lendemains + surlendemain + surlendemains:-
 /veille + veilles + vigile + vigiles + avant-veille + avant-veilles:-/week-end + week-ends:-
 /semaine + semaines:-/décade + décades:-/quinzaine + quinzaines:-/mois:-
 /trimestre + trimestres + semestre + semestres:-
 /saison + saisons: + /printemps + automne + automnes + hiver + hivers: + /été + étés: + /année + années: + /an + ans: + /siècle + siècles:-/millénaire + millénaires:-/ère + ères:-/

12.41. FICHER NTDB.TAB

seconde + secondes + tierce + tierces + milliseconde + millisecondes + microseconde + microsecondes + nanoseconde + nanosecondes + picoseconde + picosecondes:-
 /minute + minutes:-/heure + heures:-/aube + aubes + aurore + aurores:-
 /crépuscule + crépuscules + brune + brunes:-/matin + matins:-/midi:-/après-midi:-
 /tantôt + tantôts:-/soir + soirs:-/nuit + nuits + nuitée + nuitées:-/matinée + matinées:-
 /soirée + soirées:-/journée + journées:-/jour + jours + férie + fêtes:-
 /dimanche + dimanches + lundi + lundis + mardi + mardis + mercredi + mercredis + jeudi + jeudis + vendredi + vendredis + samedi + samedis:-
 /lendemain + lendemains + surlendemain + surlendemains:-
 /veille + veilles + vigile + vigiles + avant-veille + avant-veilles:-/week-end + week-ends:-
 /semaine + semaines:-/décade + décades:-/quinzaine + quinzaines:-/mois:-
 /trimestre + trimestres + semestre + semestres:-
 /saison + saisons: + /printemps + automne + automnes + hiver + hivers: + /été + étés: + /année + années: + /an + ans: + /siècle + siècles:-/millénaire + millénaires:-/ère + ères:-/

12.42. FICHER NTDEAD.TAB

seconde + secondes + tierce + tierces + milliseconde + millisecondes + microseconde + microsecondes + nanoseconde + nanosecondes + picoseconde + picosecondes:-----
 /minute + minutes:-----/heure + heures:-----/aube + aubes + aurore + aurores: + - + - -
 /crépuscule + crépuscules + brune + brunes: + - + - -/matin + matins:-----/midi:-----
 /après-midi: + - + - -/tantôt + tantôts:-----/soir + soirs:-----
 /nuit + nuits + nuitée + nuitées: + - + - -/matinée + matinées: + - + - -/soirée + soirées: + - + - -
 /journée + journées: + - + - -/jour + jours + férie + fêtes:-----
 /dimanche + dimanches + lundi + lundis + mardi + mardis + mercredi + mercredis + jeudi + jeudis + vendredi + vendredis + samedi + samedis:-----
 /lendemain + lendemains + surlendemain + surlendemains:-----
 /veille + veilles + vigile + vigiles + avant-veille + avant-veilles:-----/week-end + week-ends:-
 -----/semaine + semaines:-----/décade + décades:-----/quinzaine + quinzaines:-----
 /mois:-----/trimestre + trimestres + semestre + semestres:-----/saison + saisons:-----
 /printemps + automne + automnes + hiver + hivers:-----/été + étés:-----/année + années:-----
 ---/an + ans:-----/décennie + décennies:-----/siècle + siècles:-----
 /millénaire + millénaires:-----/ère + ères:-----/

12.43. FICHER NTDECA.TAB

seconde + secondes + tierce + tierces + milliseconde + millisecondes + microseconde + microsecondes + nanoseconde + nanosecondes + picoseconde + picosecondes:-----
 /minute + minutes:-----/heure + heures:-----
 /aube + aubes + aurore + aurores: + + + + + + + + + +-----
 /crépuscule + crépuscules + brune + brunes: + + + + + + + + + +-----
 /matin + matins: + + + + + + + + + +-----/midi: + + + + + + + + + +-----/après-midi: + + + + + + + + + +-----/tantôt + tantôts: + + + + + + + + + +-----
 /soir + soirs: + + + + + + + + + +-----
 /nuit + nuits + nuitée + nuitées: + + + + + + + + + +-----
 /matinée + matinées: + + + + + + + + + +-----/soirée + soirées: + + + + + + + + + +-----
 --/journée + journées: + + + + + + + + + +-----
 /jour + jours + férie + feries: + + + + + + + + + +-----
 /dimanche + dimanches + lundi + lundis + mardi + mardis + mercredi + mercredis + jeudi + jeudis + vendredi + vendredis + samedi + samedis:-----
 + + + /lendemain + surlendemain: + + + + + + + + + +-----
 /veille + veilles + vigile + vigiles + avant-veille + avant-veilles: + + + + + + + + + +-----
 /week-end + week-ends:----+ + + + + + + + + + /semaine + semaines:----+ + + + + + + + + +
 -+ + + /décade + décades:----+ + + + + + + + + + /quinzaine + quinzaines:----
 + + + + + + + + + + /mois:-----
 + + + /trimestre + trimestres + semestre + semestres:-----+ + + /saison + saisons:-----
 -----+ + /printemps + automne + automnes + hiver + hivers:-----+ + /été + étés:--
 -----+ + /année + années:-----/an + ans:-----
 /décennie + décennies:-----+ + /siècle + siècles:-----
 /millénaire + millénaires:-----/ère + ères:-----/

12.44. FICHER NTDENF.TAB

seconde + secondes + tierce + tierces + milliseconde + millisecondes + microseconde + microsecondes + nanoseconde + nanosecondes + picoseconde + picosecondes:-----
 /minute + minutes:-----/heure + heures:-----
 /aube + aubes + aurore + aurores: + + + + + + + /crépuscule + crépuscules + brune + brunes: + + + + + + + /matin + matins: + + + + + + + /midi: + + + + + + + /après-midi: + + + + + + + /tantôt + tantôts: + + + + + + + /soir + soirs: + + + + + + + /nuit + nuits + nuitée + nuitées: + + + + + + + /matinée + matinées: + + + + + + + /soirée + soirées: + + + + + + + /journée + journées: + + + + + + + /jour + jours + férie + feries: + + + + + + + /dimanche + dimanches + lundi + lundis + mardi + mardis + mercredi + mercredis + jeudi + jeudis + vendredi + vendredis + samedi + samedis: + + + + + + + /lendemain + lendemains + surlendemain + surlendemain: + + + + + + + /veille + veilles + vigile + vigiles + avant-veille + avant-veilles: + + + + + + + /week-end + week-ends: + + + + + + + /semaine + semaines: + + + + + + + /décade + décades: + + + + + + + /quinzaine + quinzaines: + + + + + + + /mois: + + + + + + + /trimestre + trimestres + semestre + semestres: + + + + + + + /saison + saisons:-----
 /printemps + automne + automnes + hiver + hivers:-----/été + étés:-----
 /année + années:-----/an + ans:-----/siècle:-----/siècle + siècles:-----
 /millénaire + millénaires:-----/ère + ères:-----/

seconde + secondes + tierce + tierces + milliseconde + millisecondes + microseconde + microsecondes + nanoseconde + nanosecondes + picoseconde + picosecondes:-
 /minute + minutes:-/heure + heures:-/aube + aubes + aurore + aurores:-
 /crépuscule + crépuscules + brune + brunes:-/matin + matins:-/midi:-/après-midi:-
 /tantôt + tantôts:-/soir + soirs:-/nuit + nuits + nuitée + nuitées:-/matinée + matinées:-
 /soirée + soirées:-/journée + journées:-/jour + jours + férie + feries:-
 /dimanche + dimanches + lundi + lundis + mardi + mardis + mercredi + mercredis + jeudi + jeudis + vendredi + vendredis + samedi + samedis:-
 /lendemain + lendemains + surlendemain + surlendemains:-
 /veille + veilles + vigile + vigiles + avant-veille + avant-veilles:-/week-end + week-ends:-
 /semaine + semaines:-/décade + décades:-/quinzaine + quinzaines:-/mois:-
 /trimestre + trimestres + semestre + semestres:-
 /saison + saisons: + /printemps + automne + automnes + hiver + hivers: + /été + étés: + /année + années: + /an + ans: + /siècle + siècles:-/millénaire + millénaires:-/ère + ères:-/

12.47. FICHER NTER.TAB

seconde + secondes + tierce + tierces + milliseconde + millisecondes + microseconde + microsecondes + nanoseconde + nanosecondes + picoseconde + picosecondes:-
 /minute + minutes:-/heure + heures:-/aube + aubes + aurore + aurores:-
 /crépuscule + crépuscules + brune + brunes:-/matin + matins:-/midi:-/après-midi:-
 /tantôt + tantôts:-/soir + soirs:-/nuit + nuits + nuitée + nuitées:-/matinée + matinées:-
 /soirée + soirées:-/journée + journées:-/jour + jours + férie + feries:-
 /dimanche + dimanches + lundi + lundis + mardi + mardis + mercredi + mercredis + jeudi + jeudis + vendredi + vendredis + samedi + samedis:-
 /lendemain + lendemains + surlendemain + surlendemains:-
 /veille + veilles + vigile + vigiles + avant-veille + avant-veilles:-/week-end + week-ends:-
 /semaine + semaines:-/décade + décades:-/quinzaine + quinzaines:-/mois:-
 /trimestre + trimestres + semestre + semestres:-/saison + saisons:-
 /printemps + automne + automnes + hiver + hivers:-/été + étés:-
 /année + années: + /an + ans: + /décennie + décennies: + /siècle + siècles: + /millénaire + millénaires: + /ère + ères:-/

12.48. FICHER NTMO.TAB

seconde + secondes + tierce + tierces + milliseconde + millisecondes + microseconde + microsecondes + nanoseconde + nanosecondes + picoseconde + picosecondes: + + + + + - + - - - -
 + + + + + + + - /minute + minutes: + + + + + - + - - - - + + + + + + + + -
 /heure + heures: + + + + + - + - - - - + + + + + + + + -
 /aube + aubes + aurore + aurores: + + + + + - + - - - - + + + + + + + + -
 /crépuscule + crépuscules + brune + brunes: + + + + + - + - - - - + + + + + + + + -
 /matin + matins: + + + + + - + - - - - + + + + + + + + - /midi: + + + + + - + - - - - + + + + + + + + -
 /après-midi: + + + + + - + - - - - + + + + + + + + - /tantôt + tantôts: + + + + + - + - - - - + + + + + + + + -
 /soir + soirs: + + + + + - + - - - - + + + + + + + + -
 /nuit + nuits + nuitée + nuitées: + + + + + + + - - - - + + + + + + + + -
 /matinée + matinées: + + + + + - + - - - - + + + + + + + + - /soirée + soirées: + + + + + - + - - - -
 + + + + + + + - /journée + journées: + + + + + - + - - - - + + + + + + + + -
 /jour + jours + férie + feries: + + + + + - + - - - - + + + + + + + + -
 /dimanche + dimanches + lundi + lundis + mardi + mardis + mercredi + mercredis + jeudi + jeudis + vendredi + vendredis + samedi + samedis: + + + + + - + - - - - + + + + + + + + -
 /lendemain + lendemains + surlendemain + surlendemains: - + + + + - - - - - + + + + + + + + -
 /veille + veilles + vigile + vigiles + avant-veille + avant-veilles: - + + + + - - - - - + + + + + + + + -
 /week-end + week-ends: + + + + + + + - - - - + + + + + + + + -
 /semaine + semaines: + + + + + + + - - - - + + + + + + + + - /décade + décades: + + + + + + + -
 - - - + + + + + + + - /quinzaine + quinzaines: + + + + + + + - - - - + + + + + + + + -
 /mois: + + + + + + + - - - - + + + + + + + + -
 /trimestre + trimestres + semestre + semestres: + + + + + + + - - - - + + + + + + + + -
 /saison + saisons: + + + + + + + - - - - + + + + + + + + -
 /printemps + automne + automnes + hiver + hivers: + + + + + + + - - - - + + + + + + + + -
 /été + étés: + + + + + + + - - - - + + + + + + + + - /année + années: + + + + + + + - - - -
 + + + + + + + + /an + ans: + + + + + + + - - - -
 + + + + + + + + /décennie + décennies: + + + + + + + - - - -
 + + + + + + + + /siècle + siècles: + + + + + + + - - - -
 + + + + + + + + /millénaire + millénaires: + + + + + + + - - - -
 + + + + + + + + /ère + ères: + + + + + + + - - - - + + + + + + + + + /

12.49. FICHER NTNT.TAB

```

seconde + secondes + tierce + tierces + milliseconde + millisecondes + microseconde + micr
osecondes + nanoseconde + nanosecondes + picoseconde + picosecondes:----- + ----
-----/minute + minutes:----- + -----/heure + heures:----- + -----
-----/aube + aubes + aurore + aurores:----- + -----
/crépuscule + crépuscules + brune + brunes:----- + -----/matin + matins:-----
----- + -----/midi:----- + -----/après-midi:----- + -----
-----/tantôt + tantôts:----- + -----/soir + soirs:----- + -----
/nuit + nuits + nuitée + nuitées:----- + -----/matinée + matinées:-----
+ -----/soirée + soirées:----- + -----/journée + journées:-----
- + -----/jour + jours + férie + feries:----- + -----
/dimanche + dimanches + lundi + lundis + mardi + mardis + mercredi + mercredis + jeudi + j
eudis + vendredi + vendredis + samedi + samedis:---- + + + + +---- + + + +-----
/lendemain + lendemains + surlendemain + surlendemain:---- + + + + +---- + + + +-----
-----/veille + veilles + vigile + vigiles + avant-veille + veilles:---- + + + + +---- + -- +-----
---/week-end + week-ends:----- + + -----/semaine + semaines:-----
-----/décade + décades:-----/quinzaine + quinzaines:-----
-----/mois:-----
/trimestre + trimestres + semestre + semestres:-----/saison + saisons:-
-----/printemps + automne + automnes + hiver + hivers:-----
-----/été + étés:-----/année + années:-----
/an + ans:-----/décennie + décennies:-----
/siècle + siècles:-----/millénaire + millénaires:-----
---/ère + ères:-----/

```

seconde + secondes + tierce + tierces + milliseconde + millisecondes + microseconde + microssecondes + nanoseconde + nanosecondes + picoseconde + picosecondes: + + + + + + +
+ + + + + + + + -
+ + + + + /minute + minutes: + + + + + + + + + + + + + + + + + -
+ + + + + /heure + heures: + + + + + + + + + + + + + + + + + -
+ + + + + /aube + aubes + aurore + aurores: + + + + + + + + + + + + + + + + + -
+ + + + + /crépuscule + crépuscules + brune + brunes: + + + + + + + + + + + + + + + + +
+ + - + + + + /matin + matins: + + + + + + + + + + + + + + + + + -
+ + + + + /midi: + + + + + + + + + + + + + + + + + - + + + + + /après-midi: + + + + + + + + + + + + + + + + + -
+ + + + + /tantôt + tantôts: + + + + + + + + + + + + + + + + + -
+ + + + + /soir + soirs: + + + + + + + + + + + + + + + + + -
+ + + + + /nuit + nuits + nuitée + nuitées: + + + + + + + + + + + + + + + + + -
+ + + + + /matinée + matinées: + + + + + + + + + + + + + + + + + -
+ + + + + /soirée + soirées: + + + + + + + + + + + + + + + + + -
+ + + + + /journée + journées: + + + + + + + + + + + + + + + + + -
+ + + + + /jour + jours + férie + fêtes: + + + + + + + + + + + + + + + + + -
+ + + + + /dimanche + dimanches + lundi + lundis + mardi + mardis + mercredi + mercredis
+ jeudi + jeudis + vendredi + vendredis + samedi + samedis: + + + + + + + + + + + + + + + + +
+ + + + + -
+ + + + + /lendemain + lendemains + surlendemain + surlendemains: + + + + + + + + + +
+ + + + + + + + + + - + + + + + /veille + veilles + vigile + vigiles + avant-veille + avant-veilles:
+ - + + + + + /week-end + week-ends: + -
+ + + + + /semaine + semaines: + -
+ + + + + /décade + décades: + -
+ + + + + /quinzaine + quinzaines: + -
+ + + + + /mois: + -
+ + + + + /trimestre + trimestres + semestre + semestres: + + + + + + + + + + + + + + + + + +
+ + + + + - + + + + + /saison + saisons: + -
+ + + + + /printemps + automne + automnes + hiver + hivers: + + + + + + + + + + + + + + + + +
+ + + + + + - + + + + + /été + étés: + -
+ + + + + /année + années: + -
+ + + + + /an + ans: + -
+ + + + + /siècle: + -
+ + + + + /siècle + siècles: + -
+ + + + + /millénaire + millénaires: + -
+ + + + + /ère + ères: + - + + + + + /

à peine:+++++/à peu près+à
 peu de chose près:+++++/au
 mieux:+++++/au
 moins:+++++/au
 pire:+++++/au
 pis:+++++/au plus+ tout au
 plus:+++++/à tout
 casser:+++++/en
 gros:+++++/en tout+ en tout
 et pour
 tout:+++++/approximative
 nt:+++++/exactement:+++
 ++++/pratiquement:++++
 ++++/précisément:++++
 ++++/seulement:++++
 ++++/simplement:++++
 ++++/uniquement:++++
 ++++/avant
 tout:+++++/pas tout à
 fait:+++++/surtout:++++
 ++++/tout+ toute+ tous+ toutes:+++
 ++++/ensuite:++++
 ++++/environ:++++
 ++++/même:++++
 ++++/plutôt:++++
 ++++/presque:++++/

à peine:+++++/à peu près+/à
peu de chose près:+++++/au
mieux:+++++/au
moins:+++++/au
pire:+++++/au
pis:+++++/au plus+tout au
plus:+++++/à tout
casser:+++++/en
gros:+++++/en tout+en tout
et pour
tout:+++++/approximative
ment:+++++/exactement:+++
+++++/pratiquement:++++
+++++/précisément:++++
+++++/seulement:++++
+++++/simplement:++++
+++++/uniquement:++++
+++++/avant
tout:+++++/pas tout à
fait:+++++/surtout:++++
+++++/tout+toute+tous+toutes:++
+++++/ensuite:++++
+++++/environ:++++
+++++/même:++++
+++++/plutôt:++++
+/presque:++++/

[illegible][illegible]

12.57. FICHIER PDDT.TAB

à peine:+++-+---
 ++++++/à peu près+à peu
 de chose près:++-+----
 ++++++/au
 mieux:+++++
 ++++/au moins:++-+---
 ++++++/au
 pire:+++++
 ++/au
 pis:+++++
 +/au plus+ tout au plus:++-+---
 ++++++/à tout
 casser:+++++
 ++++/en
 gros:+++++
 ++/en tout+en tout et pour
 tout:+++++
 ++/approximativement:+++++
 ++++/exactement:++-+----
 ++++++/pratiquement:++
 ++++++/préc
 isément:+++++
 ++++/seulement:+++++
 ++++++/simplement:+++++
 ++++++/uniquement:+++++
 ++++++/avant
 tout:+++++
 ++/pas tout à fait:++-+----
 ++++++/surtout:+++++
 ++++++/tout+ tout
 e+ tous+ toutes:-----+---+---
 ++++++/ensuite:+++++
 ++++++/environ:++-+---
 ++++++/même:+++++
 ++++++/plutôt:++
 ++++++/pres
 que:++-+---+

à peine: ++++++/à peu près: +++++/
peu de chose près: +++++/+
mieux: ++++++/au
moins: ++++++/au
pire: ++++++/au
pis: ++++++/au plus + tout au
plus: ++++++/à tout
casser: ++++++/en
gros: ++++++/en tout + en tout
et pour
tout: ++++++/approximativement: +++++/
nt: ++++++/exactement: ++++
+++++/pratiquement: ++++++
+++++/précisément: ++++++
+++++/seulement: ++++++
+++++/simplement: ++++++
+++++/uniquement: ++++++
+++++/avant
tout: ++++++/pas tout à
fait: ++++++/surtout: ++++
+++++/tout + toute + tous + toutes: ++++
+++++/ensuite: ++++++
+++++/environ: ++++++
+++++/même: ++++++
+++++/plutôt: ++++++
+/presque: ++++++/

12.60. FICHER PDLSNT.TAB

à peine:+++++++/à peu près+à
 peu de chose près:+++++++/au
 mieux:+++++++/au
 moins:+++++++/au
 pire:+++++++/au
 pis:+++++++/au plus+ tout au
 plus:+++++++/à tout
 casser:+++++++/en
 gros:+++++++/en tout+ en tout
 et pour
 tout:+++++++/approximativeme
 nt:+++++++/exactement:+++
 ++++++++/pratiquement:+++++
 ++++++++/précisément:+++++
 ++++++++/seulement:+++++
 ++++++++/simplement:+++++
 ++++++++/uniquement:+++++
 ++++++++/avant
 tout:+++++++/pas tout à
 fait:+++++++/surtout:+++++
 ++++++++/tout+ toute+ tous+ toutes:+++
 ++++++++/ensuite:+++++
 ++++++++/environ:+++++
 ++++++++/même:+++++
 ++++++++/plutôt:+++++
 ++++++++/presque:+++++

12.61. FICHER PDNT.TAB

à peine:-----+-----/à peu près+à peu de chose près:-----+-----
 -----/au mieux:-----+-----/au moins:-----+-----/au
 pire:-----+-----/au pis:-----+-----/au plus+ tout au plus:--
 -----+-----/à tout casser:-----+-----/en gros:-----+--
 -----/en tout+ en tout et pour tout:-----+-----
 /approximativement:-----+-----/exactement:-----+-----
 /pratiquement:-----+-----/précisément:-----+-----
 /seulement:-----+-----/simplement:-----+-----
 /uniquement:-----+-----/avant tout:-----+-----/pas tout
 à fait:-----+-----/surtout:-----+-----
 /tout+ toute+ tous+ toutes:-----+-----/ensuite:-----+-----
 /environ:-----+-----/même:-----+-----/plutôt:-----
 +-----/presque:-----+-----/

[illegible]

12.68. FICHIER *PRCSDNNT.TAB*

à:-----+----+++++---++++/à partir
 de:+++++ /après:+++++
 +++++ /aussitôt+sitôt:+++++
 +++++ /avant:+++++
 +++++ /dans:+++++
 +++++ /depuis:+++++
 + /dès:+++++ /d'ici:+++++
 +++++ /d'ici
 à:+++++ /durant:+++++
 +++++ /en:-----
 /jusque:+++++ /jusqu'à:++
 +++++ /jusqu'en:-----
 ----- /jusque vers:+++-----+----+---
 +++++ /passé:+++++ /pe
 ndant:+++++ /pour:+++++
 +-+++++
 +++++ /sous:+++++
 +++++ /sur:+++++ /ver
 s:+++-----+----+++++---+++++ /

12.69. FICHIER *PRCSNT.TAB*

à:-----+----+++++---++++/à partir
 de:+++++ /après:+++++
 +++++ /aussitôt+sitôt:+++++
 +++++ /avant:+++++
 +++++ /dans:+++++
 +++++ /depuis:+++++
 + /dès:+++++ /d'ici:+++++
 +++++ /d'ici
 à:+++++ /durant:+++++
 +++++ /en:+++++
 +++++ /jusque:+++++
 +++++ /jusqu'à:+++++
 +++++ /jusqu'en:+++++
 +++++ /jusque vers:+++-----+----
 +++++
 +++++ /passé:+++++ /pe
 ndant:+++++ /pour:+++++
 +-+++++
 +++++ /sous:+++++
 +++++ /sur:+++++ /ver
 s:+++-----+----+++++---+++++ /

à+aux:+++++
de:+++++ /après:+++++
+++++ /aussitôt+sitôt:+++++
+++++ /avant:+++++
+++++ /dans:+++++--
+++++ /depuis:+++++
+++++ /dès:+++++ /d'ici:+++++
ci:+++++ /d'ici à+d'ici
aux:+++++ /durant:+++++
+++++ /en:+++++
+++++ /jusque:-----
/jusqu'à+ jusqu'aux:-----/jusqu'en:-----/jusque
vers:-----
/passé:+++++ /pendant:+++++
+++++ /pour:+++++
+++++ /sous:+++++
+++++ /sur:+++++ /vers:-----
-----/

à + au:-----/à partir de:-----
/après:+++++++/aussitôt + sitôt:

/avant:+++++++/dans:++ +-----
-- + --
+++++++/depuis:++++++
+++++++/dès:-----
/d'ici:+++++++/d'ici à + d'ici
au:-----
/durant:+++++++/en:-----
-----/jusque:-----/jusqu'à + jusqu'au:-----
-----/jusqu'en:-----/jusque vers:-----
/passé:+++++++/pendant:++ +
+++++++/pour:++++++
+++++++/sous:-----
/sur:+++++++/vers:-----
-----/

à+au+aux:++++++-----+---+-+++/à partir
de:+++++++--+++++-----+---+-++-
++/après:+++++++-----+---+-++-
+++++/-+/aussitôt+sitôt:-----+---+-----+---+-
++/avant:+++++++-----+---+-++-
+++++/-+/dans:+++++++--++++++-----+---+-
+++++-----
/depuis:+++++++-----+---+-++-
++/dès:-----+---+-++-
++/d'ici:+++++++--++++++-----+---+-
++/d'ici à+d'ici au+d'ici aux:+++++++--++++++-----+---+-
+++++-----
++/durant:+++++++-----+---+-++-
+++++/-+/en:+++++++-----+---+-++-
/jusque:+++++++--++++++-----+---+-
/jusqu'à+jusqu'au+jusqu'aux:+++++++-----
+++++-----+---+-++-/jusqu'en:+++++++-----
-----+---+-++-/jusque vers:+++++++-----+---+-
+++++-----
++/passé:+++++++-----+---+-++-
+++++/-+/pendant:+++++++-----
+++++-----+---+-++-
++/pour:+++++++-----+---+-++-
+++/sous:+++++++-----/sur:+++++++-----
+++++-----+---+-++-/vers:+++++++-----+---+-
---++-----+---+-+++/

à + au + aux: + + - + + - /à partir de: + + - + + - /après: + + - + + -
/aussitôt + sitôt: + + - + + - /avant: + + - + + - /dans: ----- /depuis: + + - + + -
/dès: + + - + + - /d'ici: + + - + + - /d'ici à + d'ici au + d'ici aux: + + - + + -
/durant: + + - + + - /en: ----- /jusque: ----- /jusqu'à + jusqu'au + jusqu'aux: + +
+ + + - /jusqu'en: ----- /jusque vers: + + - + + - /passé: + + - + + - /pendant: + +
+ + + - /pour: + + - + + - /sous: ----- /sur: ----- /vers: + + - + + -

[illegible][illegible]

à + au + aux:-----/à partir de:-----/après:-----
-----/aussitôt + sitôt:-----/avant:-----
--/dans:-----/depuis:-----/dès:-----
-----/d'ici:-----/d'ici à + d'ici au + d'ici aux:-----
---/durant:-----/en:+++-+--++++++-----+++++-----
--/jusque:-----/jusqu'à + jusqu'au + jusqu'aux:-----
--/jusqu'en:-----+++-----+-+-----/jusque vers:-----
/par:++++++-++++++-+++++++/passé:-----
-----/pendant:-----/pour:-----/sous:-----
-----/sur:-----/vers:-----/

13. EXEMPLES.

Cette précocité tient d'abord à une régression... de la soumission millénaire à l'ordre divin.(Journal Le Monde du 23/8/86)
 les adversaires de la «perestroïka» sont, depuis dix jours, quotidiennement dénoncés...(Journal Le Monde du 16/4/88)
 53% des décès concernaient des enfants âgés de moins de cinq ans(Journal Le Monde du 19/2/87)
Trois enfants âgés de moins de douze ans sur un total de 281 n'ayant pas encore quinze ans sont...(Journal Le Monde du 13/2/87)
Cinq ans seulement après les premiers cas...(Journal Le Monde du 7/2/87)
Deux ans après son arrivée au pouvoir, le 11 mars 1985, M. Gorbatchev...(Journal Le Monde du 11/3/87)
 Ayant accompli -au prix, à ce jour, de cent cinquante morts- l'exploit de «rapatrier l'affrontement» israélo-arabe vers leur terre...(Journal Le Monde du 22/4/88)
 Cela n'a pas d'importance que cela prenne dix, vingt, trente ans ou plus...(Journal Le Monde du 7/8/86)
 Il a fait défection le 7 novembre 1981, un mois et une semaine avant la proclamation...(Journal Le Monde du 17/4/87)
 le marché est différent aujourd'hui de ce qu'il était en 1973, voire en 1979-1980...(Journal Le Monde du 8/8/87)
 Le nombre des sans-emplois... a augmenté de 54% entre janvier et novembre l'an dernier...(Journal Le Monde du 5/8/86)
 arrêté dans la nuit du lundi 21 au mardi 22 juillet à Arbonne...(Journal Le Monde du 30/7/86)
 les trois derniers rois carolingiens, qui régnèrent de 936 à 987, ne furent pas... Mais, au début des années 80, le Carolingien...(Journal Le Monde du 22/5/87)
Quelques semaines après le début du soulèvement, les leaders... firent connaître une plate-forme revendicative...(Journal Le Monde du 22/4/88)
2340 sont arrivés au cours du seul mois de juillet en RFA...(Journal Le Monde du 11/8/86)
A deux mois environ du sixième anniversaire de la guerre irano-irakienne, que lui-même a déclenchée le 22 septembre 1980, le président Saddam Hussein a appelé le 2 août les dirigeants iraniens...(Journal Le Monde du 5/8/86)
Dans quelques jours, le 3 septembre, l'Afrique du Sud entrera dans sa troisième année de tourmente...(Journal Le Monde du 23/8/86)
 Il s'agit là sans doute de l'incident le plus sanglant à Lhassa depuis le soulèvement de 1959..., mais aussi du plus grave ces dernières années dans une des régions frontalières... Qu'un tel incident se produise... le jour de la "libération"..., et ce à moins d'un mois du treizième congrès du PCC, représente certainement un coup dur pour le régime.(Journal Le Monde du 5/10/87)
 les grandes décisions... sont prises à l'issue du Congrès populaire général qui se tient tous les ans, vers la fin du mois de février, avec la participation...(Journal Le Monde du 25/3/87)
Un homme d'affaires japonais faisait remarquer l'autre jour... à quel point les Français mènent une belle vie.(Journal Le Monde du 9/9/87)
 Cet acquis pour une population civile qui souffre depuis tant d'années est la conséquence...(Journal Le Monde du 28/3/88)
 commencé le 16, le vingtième congrès... devait se terminer... samedi 18 avril...(Journal Le Monde du 18/4/87)
Dans la nuit du 12 octobre 1984, aussitôt après l'attentat...(Journal Le Monde du 15/6/87)
 en organisant à partir du 4 septembre une semaine consacrée à...(Journal Le Monde du 23/9/86)
 La campagne..., menée deux dimanches consécutifs, s'est... essoufflée...(Journal Le Monde du 28/3/87)
 la grève des PTT organisée le 11 par les syndicats... nul doute que le vote du 12 n'encourage...(Journal Le Monde du 15/6/87)
 La première audience avait eu lieu le 10 février: six mois, à peu près la mi-parcours, c'est en effet la bonne distance pour tenter un bilan.(Journal Le Monde du 16/8/86)
Le 14 septembre, quarante-huit heures après un attentat...(Journal Le Monde du 16/9/86)
le premier secrétaire du Parti socialiste conserve l'acquis de mars avec quatre siège,...(Journal Le Monde du 30/9/86)
 Le record du mois d'août de l'année dernière (9178) a été battu au mois de juillet (9710). Les chiffres des premiers mois de l'année semblent indiquer...(Journal Le Monde du 11/8/86)
 Depuis l'entrée des troupes syriennes le 22 février, vingt-deux explosions... ont été dénombrées... l'accélération du processus est nette. Rien jusqu'au 8 mars, mais onze explosions depuis le début de la semaine, qui commencent...(Journal Le Monde du 28/3/87)

Lundi soir 9 février, à cinq minutes de l'ultimatum qu'elle avait fixé (minuit), l'organisation... Elle avait menacé de les tuer si, à cette heure, quatre cents Palestiniens...(Journal Le Monde du 6/2/87)

Le 5 août a sans doute été un jour comme les autres pour Nelson Mandela. Le 9131e en détention,..., soit vingt-cinq années de sa vie. C'est en effet le 5 août 1962 que Nelson Mandela a été arrêté...(Journal Le Monde du 6/8/87)

Près de deux mois et demi après la déclaration...(Journal Le Monde du 23/8/86)

une toute petite partie du mobilier funéraire découvert en 1939, 1940 et 1946 par Pierre Montet... Peut-être souffrent-ils du voisinage... avec ceux de Toutankhamon (environ 1354-1346 avant Jésus-Christ). Mais surtout, leur découverte pendant et juste après la guerre de 1939-1945 n'a pas eu le retentissement... l'époque de Tanis... qui a commencé à la fin du nouvel empire (1069 avant Jésus-Christ) et s'est achevée au début de la période de la renaissance saïte (663 avant Jésus-Christ). En 1069 avant Jésus-Christ... la puissance royale était en pleine décadence depuis une bonne cinquantaine d'années... C'est d'ailleurs un Thébain... qui devient roi à Tanis sous le nom de Psousennès Ier en 1040... Pendant les règnes de Psousennès Ier (1040-993) et d'Amonémopé (993-981), l'Egypte est tranquille... Dans le même temps, Saül (vers 1030-1010) devient le premier roi des Hébreux, et son successeur David (vers 1010-975) prend Jérusalem... En 925, Chechang Ier envahit la Palestine... Le site de Tanis est, en effet, identifié depuis 1722... d'énormes débris... avaient servi de carrières de pierre sous les Omeyyades (661-750) et les Fatimides (973-1171 en Egypte). Tous les ans depuis 1964... le site de Tanis est fouillé... en témoignent de pauvres tombes... identifiées en 1985 et 1986...(Journal Le Monde du 27/3/87)

des incidents sanglants... ont eu lieu le jeudi 1er octobre - jour de la fête nationale - à Lhassa...(Journal Le Monde du 5/10/87)

des consultations qui ont lieu régulièrement deux fois par an depuis 1982 entre Chinois et Soviétiques...(Journal Le Monde du 21/8/86)

Au bazar-arsenal de Darra, à une heure de Peshawar...(Journal Le Monde du 26/9/86)

Ces idées sont développées... à la télévision aussi, deux à trois fois par semaine, à une heure de grande écoute...(Journal Le Monde du 20/8/86)

Israël vit une fois de plus à l'heure des grandes manoeuvres diplomatiques...(Journal Le Monde du 2/5/87)

ne peuvent qu'ajouter aux tensions présentes et retarder l'heure des réconciliations nécessaires...(Journal Le Monde du 19/2/87)

Hier vilipendée à Madrid et encensée à Bilbao...(Journal Le Monde du 11/8/86)

A l'âge d'or des trains japonais, du début du siècle à la deuxième guerre mondiale, le réseau...(Journal Le Monde du 3/4/87)

Pendant huit cent soixante et un ans, desquels il faut soustraire l'interruption..., de 1792 à 1814, et celle des Cent-Jours de 1815... quinze rois ont régné pendant trois cent trente-neuf ans...(Journal Le Monde du 6/4/87)

« Ce sera une guerre de cent ans », prédit un intellectuel d'Addis-Abeba...(Journal Le Monde du 5/10/87)

une main anonyme et irrévérencieuse a tracé cette inscription: "Cette année, le 1er avril tombe le 6 mai." Les élections législatives blanches... ne sont-elles donc qu'une farce?(Journal Le Monde du 30/4/87)

l'effort d'interprétation... des textes islamiques... interrompu... à partir du quatrième siècle de l'hégire (aux alentours de l'an 1000)...(Journal Le Monde du 7/8/87)

il ne verra pas le jour avant cinq ans...(Journal Le Monde du 7/2/87)

les choses se présentent depuis le début du mois de décembre sous un jour tout à fait différent...(Journal Le Monde du 4/2/87)

Au lendemain de la victoire communiste de 1975, une visite...(Journal Le Monde du 8/8/86)

De 1983 à fin 1986, plus de 2000 emplois ont été supprimés... Du début 1987 à mi-1988, on prévoit encore 390 suppressions... Parallèlement, l'indice de travail nécessaire pour construire un même navire est tombé de 91 en 1983 (indice 100 en 1976) à 81 en 1986 (on escompte 66 en 1990 dans le plan d'entreprise)...(Journal Le Monde du 28/7/87)

A la fin de ces quatre mois et demi de session ininterrompue,...(Journal Le Monde du 15/8/86)

la Baltique... où, à l'extrême fin du XVIIIème siècle, s'affrontèrent...(Journal Le Monde du 22/4/87)

Le changement... en celui d'une société nationale? Remis à l'après-1988...(Journal Le Monde du 3/2/87)

le référendum... (avant la fin de l'année) et les élections... (au printemps)...(Journal Le Monde du 15/9/86)

Les auditions publiques et télévisées... ont pris fin le lundi 3 août...(Journal Le Monde du 5/8/87)

M. Bazargan avait, vers la fin d'avril 1986, repris...(Journal Le Monde du 22/9/86)

la vitesse du progrès technologique qui, à l'orée des années 80, a permis...(Journal Le Monde du 5/8/86)

pour "boucler les fins de mois" de son pays en recevant...(Journal Le Monde du 18/2/87)

cette question a été, jusqu'à la dernière minute, le principal obstacle...(Journal Le Monde du 21/4/87)

Après la faconde des quatre années écoulées, un tel silence impressionne...(Journal Le Monde du 8/8/86)

Ce jour-là, treize ans après la déposition du négus - le 12 septembre 1974 - tous les dirigeants éthiopiens sont réunis...(Journal Le Monde du 5/10/87)

dans les heures qui avaient précédé son assassinat à la fin de l'été 1982. (Journal Le Monde du 16/8/86)

Dans les heures qui ont précédé l'annonce de... (Journal Le Monde du 13/9/86)

Des commandos d'Etreras devaient tomber dans les prochaines heures... (Journal Le Monde du 5/10/87)

elles pourraient diminuer de 45% d'ici à l'an 2000, par une amélioration... (Journal Le Monde du 5/8/86)

Il existe donc de fortes chances que, dans la guerre du golfe, la septième année, qui a commencé ce lundi 22 septembre, ne soit pas l'année décisive dont... (Journal Le Monde du 23/9/86)

La limousine... s'arrête à 16 heures pile devant... Si je pouvais ... je dormirais pendant trois ou quatre ans d'affilée... Nous avons marché... neuf jours entiers pour revenir ici. (Journal Le Monde du 26/9/86)

la solidarité dont toute la société civile a fait preuve en ces heures dramatiques. (Journal Le Monde du 21/4/87)

Le principe... a été retenu avant même le 16 mars. (Journal Le Monde du 8/8/86)

Le rapport d'Amnesty International pour l'année 1986, publié mercredi 30 septembre, c'est l'histoire de ces destins meurtris... (Journal Le Monde du 1/10/87)

les travaux ont montré que dans les cinquante ans qui avaient précédé, la durée de vie avait progressé de six ans. (Journal Le Monde du 23/8/86)

la fête de la mi-automne célébrée, dans le calendrier lunaire des Asiatiques, le quinzième jour du huitième mois, c'est-à-dire le 18 septembre. En ce jour, les festivités... (Journal Le Monde du 22/9/86)

Samedi 12 septembre au matin. (Journal Le Monde du 15/9/87)

la petite équipe... a, dans l'après-midi de vendredi 5 septembre, ramené... Pour eux... , le premier jour de la nouvelle année musulmane ne pouvait mieux commencer. (Journal Le Monde du 8/9/86)

A la mort d'Hugues le Grand (956)... Comme le rappelle Laurent Theis (4)... (Journal Le Monde du 6/4/87)

Un superbe... édit... co-signé par... en l'an moharem 1210 (1792), accorde à... (Journal Le Monde du 11/9/86)

après avoir purgé huit mois de prison sur les dix ans auxquels ils avaient été condamnés pour leur participation... , en juillet 1985... (Journal Le Monde du 30/7/86)

Si le Mexique, qui a eu la plus forte croissance démographique pendant un quart de siècle jusque 1970... y est parvenu... (Journal Le Monde du 23/8/86)

le cliché montre un «propriétaire foncier» passant un mauvais quart d'heure peu après la fondation de la Chine populaire. (Journal Le Monde du 9/4/88)

A cette époque de l'année, l'image est d'autant plus frappante qu'on voit des fonctionnaires... se promener nonchalamment en début d'après-midi dans les jardins... (Journal Le Monde du 9/9/87)

Elles ont augmenté de 43% en valeur pour les quatre premiers mois de l'année par rapport à la même période de l'an dernier. (Journal Le Monde du 5/8/86)

L'après-guerre a ouvert une parenthèse d'une vingtaine d'années, une sorte de... (Journal Le Monde du 23/8/86)

la production industrielle a augmenté, durant le premier semestre, de 12% par rapport à la période correspondante de 1985... (Sélection hebdomadaire du journal Le Monde du 5/8/88)

N'y figure pas notamment ceux qui ont été emprisonnés pour une durée inférieure à trente jours ni ceux qui ont été arrêtés au cours du dernier mois précédant le jour de la publication. (Journal Le Monde du 23/8/86)

Sur la période 1979-1983 la France... (Journal Le Monde du 5/8/86)

Dès sa première journée de travail, le forum a décidé de... (Journal Le Monde du 12/8/86)

M. Barco a provoqué une surprise en se montrant au cours de ses deux mois d'attente - et de mutisme - plus indépendant... (Journal Le Monde du 8/8/86)

à la veille de l'ouverture, le 20, du Conseil... (Journal Le Monde du 21/4/87)

Catholiques et protestants avaient annoncé que l'été verrait l'aboutissement de neuf mois de campagne... (Journal Le Monde du 23/8/86)

Juillet est le mois des parades unionistes. (Journal Le Monde du 23/8/86)

Pendant des années, les jeunes... ont vu leurs parents... subir mille humiliations... (Journal Le Monde du 22/4/88)

La Pravda... maintenant tourne le dos aux conservateurs - provisoirement peut-être mais pour l'heure totalement. (Journal Le Monde du 22/4/88)

La grève aux chantiers navals Lénine de Gdansk, qui est entrée dans sa deuxième semaine, était toujours dans l'impasse lundi 10 mai, après la rupture dans la nuit d'une troisième série de négociations... (Journal Le Monde du 10/5/88)

M. Kazimierz Barcikowski... a présenté des excuses au cardinal Glemp samedi 7 mai pour l'assaut donné par les forces de l'ordre jeudi 5 mai aux aciéries de Nowa-Huta... (Journal Le Monde du 10/5/88)

Au centre de ces manoeuvres, un vieillard... Janos Kadar... secrétaire du Parti socialiste ouvrier hongrois depuis bientôt trente-deux ans. (Journal Le Monde du 3/5/88)

Le comité central lui-même, réuni en plenum les 23 et 24 mars à moins de deux mois de la conférence nationale du parti - la première depuis 1957, - n'a pu que déplorer... (Journal Le Monde du 3/5/88)

Quelques jours plus tard, le comité central publie un «projet de prise de position», document de travail pour la conférence nationale qui s'ouvre le 20 mai. (Journal Le Monde du 3/5/88)

elles ont carrément perdu 12% de leurs membres dans la seule année de 1986. (Journal Le Monde du 3/5/88)

M. Kadar, que la rumeur disait prêt à passer la main il y a quelques mois, est tout à coup omniprésent...(Journal Le Monde du 3/5/88)

Il ne l'est plus depuis plusieurs années et il le sera encore moins à l'avenir...(Journal Le Monde du 4/5/88)

Par deux votes successifs (le 28 et le 29 avril), celle-ci a restreint considérablement... la marge...(Journal Le Monde du 4/5/88)

la Petrobas... le faisait depuis une dizaine d'années...(Journal Le Monde du 4/5/88)

En un mois (de mars à avril), le taux de chômage a diminué de 0,2% contre une réduction moyenne de 0,1% au cours des mois précédents. Au total, plus d'un million d'emplois ont été créés aux Etats-Unis depuis le début de cette année... Le regain concerne en particulier des régions touchées ces dernières années par une crise sévère...(Journal Le Monde du 9/5/88)

Les crédits à la consommation ont continué d'augmenter rapidement... au cours des trois premiers mois de l'année...(Journal Le Monde du 9/5/88)

Sagesse bien improbable en année électorale...(Journal Le Monde du 9/5/88)

l'annonce, tôt vendredi matin, de la nouvelle contraction du chômage n'a été suivie que d'une baisse très modérée de l'indice Dow Jones, qui se maintient au-dessus de 2000 depuis maintenant plusieurs semaines...(Journal Le Monde du 9/5/88)

les Etats-Unis en sont actuellement à leur soixante-cinquième mois de croissance ininterrompue...(Journal Le Monde du 9/5/88)

les députés travaillistes... finiront par obtenir un débat extraordinaire sur cette réforme pour le lendemain...(Journal Le Monde du 4/5/88)

un côté «enrichissez-vous» très dix-neuvième siècle. Là réside l'efficacité économique selon Mme Thatcher...(Journal Le Monde du 4/5/88)

le week-end en fut la démonstration, avec deux manifestations...(Journal Le Monde du 26/1/88)

Près de deux mois après la mort prématurée de Mgr Kevin McNamara...(Sélection hebdomadaire du Journal Le Monde du lundi 18/1/89)

Les Vietnamiens commenceront à évacuer le Cambodge -dans un délai de deux ans, et non plus de trente mois, a proposé, jeudi, M. Hun Sen- une fois un accord réalisé sur ce point...(Journal Le Monde du 23/1/88)

L'été confère au 14 juillet... un inaltérable parfum de fête. Chaque commémoration de la révolution d'Octobre nous rappelle qu'elle s'est déroulée en réalité, selon notre calendrier, en novembre, c'est-à-dire dans le mois le plus frileux, le plus triste, en un mot le moins révolutionnaire de l'année...(Journal Le Monde du 11/11/87)

Lénine avait déclaré le 7 mars précédent devant le congrès de son parti: «C'est une...»(Journal Le Monde du 11/11/87)

L'équipe au pouvoir aura d'autant plus besoin de stabilité, en ces heures difficiles, qu'elle reste confrontée à de graves défis...(Journal Le Monde du 12/11/87)

Pour les dirigeants nigériens, le péril majeur, aujourd'hui comme hier, vient du nord...(Journal Le Monde du 12/11/87)

ses troupes ont été engagées ces derniers jours dans des combats contre des forces «russo-cubaines»...(Journal Le Monde du 14/11/87)

Dans la chaleur du petit matin, la capitale sommeille...(Journal Le Monde du 16/10/87)

les trois ou cinq ans à venir ne s'annoncent pas aisé.(Journal Le Monde du 15/10/87)

Depuis le mois de juin dernier, deux autres cas de figure comparables montrent qu'une droite franchement majoritaire...(Journal Le Monde du 20/10/87)

Coup sur coup, ces deux derniers jours, le président américain... vient d'encaisser trois revers majeurs...(Journal Le Monde du 26/10/87)

Bagdad eut alors recours aux grands moyens, entreprenant à partir de mai 1987 une série d'expéditions...(Journal Le Monde du 23/10/87)

la libéralisation devrait, indiquait-on le 22 au bureau politique, assurer l'enregistrement...(Journal Le Monde du 24/10/87)

Le comité central... a approuvé, le 21 octobre, le... rapport que M. Mikhaïl Gorbatchev doit présenter le 2 novembre prochain, à l'occasion du soixante-dixième anniversaire de la révolution...(Journal Le Monde du 23/10/87)

les cours fléchirent en l'espace de quelques heures...(Journal Le Monde du 24/10/87)

Deux cents journalistes étrangers ont ainsi pu voir M. Zhao Ziyang... défendre pendant deux heures et demie la thèse de la réforme...(Journal Le Monde du 27/10/87)

La CGT a organisé, le lundi 26 octobre, une journée nationale d'action...(Journal Le Monde du 26/10/87)

Cette controverse, alimentée hier par l'affaire de la Chapelle Darblay, aujourd'hui par le procès de Nanterre, sert de révélateur...(Journal Le Monde du 26/10/87)

le chiffre de trente et un morts avancé, vendredi soir 4 mars, par l'agence Tass...(Journal Le Monde du 8/3/88)

une crise qui nuît gravement à leur image de marque...(Journal Le Monde du 8/3/88)

Une semaine de navette diplomatique... s'achève ainsi...(Journal Le Monde du 7/3/88)

la responsabilité de l'opération menée quelques heures plutôt par trois Palestiniens...(Journal Le Monde du 9/3/88)

Il est à peine 7 heures lundi 7 mars... lorsque trois Palestiniens... pénètrent en Israël. Vers 7 h 30, ils arrêtent une Renault-4...(Journal Le Monde du 9/3/88)

Il s'agit ici de rassurer les Arabes qui craignent qu'Israël ne s'engage dans un processus qui se limiterait à l'automne...(Journal Le Monde du 8/3/88)

Le premier week-end de mars... a été un des plus violents des dernières semaines...(Journal Le Monde du 8/3/88)

Ils menacent de tuer un otage toutes les heures...(Journal Le Monde du 9/3/88)

presque chaque jour des familles déménagent...(Journal Le Monde du 22/3/88)

Depuis que Staline, dès la fin des années 20, avait imposé le monolithisme au parti...(Journal Le Monde du 21/3/88)

le Saoudien transfère... les tableaux volés... un jour sur le Nabila, trois semaines plus tard au dernier étage de l'immeuble «les Floriales»... puis de nouveau, et nuitamment, sur le Nabila.(Journal Le Monde du 17/3/88)

la croissance assez remarquable des années 60 et 70 est en passe de n'être plus qu'un souvenir.(Journal Le Monde du 17/3/88)

Depuis sa désignation l'an passé, le nouveau premier ministre... a décidé de jouer le jeu...(Journal Le Monde du 17/3/88)

entre le 29 février, début de la reprise des tirs, et le dix mars, date du cessez-le-feu proposé par Bagdad, soixante-huit SCUD-B avaient atteint Téhéran...(Journal Le Monde du 23/3/88)

les offensives Kerbela menées par l'Iran fin 1986-début 1987 visaient à faire tomber Bassora...(Journal Le Monde du 23/3/88)

Rengo... qui mènera fin mars et début avril le mouvement de revendication...(Journal Le Monde du 16/3/88)

L'élément le plus «neuf» des deux journées d'entretien, les mardi et mercredi 15 et 16 mars, est sans doute le ton particulièrement ferme...(Journal Le Monde du 22/3/88)

Le même jour, une commission internationale serait créée afin d'étudier les responsabilités...(Journal Le Monde du 5/11/87)

Il a également souhaité que des élections... soient organisées dans un délai de deux mois après la promulgation du texte.(Journal Le Monde du 5/11/87)

Il a seize ans en 1944 quand se constitue, du 6 au 18 juin... l'unique maquis d'importance du Morbihan.(Journal Le Monde du 17/12/87)

les trois quarts des paysans sont à plus d'une demi-journée de marche d'une route carrossable(Journal Le Monde du 25/9/87)

l'évolution récente de la production agricole, même en année «normale», rend illusoire l'objectif officiel...(Journal Le Monde du 25/9/87)

moins d'une heure plus tard, le colonel Rabuka annonçait à la radio...(Journal Le Monde du 28/9/87)

un accord qui semblait, jeudi encore, en passe d'être conclu...(Journal Le Monde du 28/9/87)

Au cours des dernières semaines, divers signes de tension s'étaient fait jour...(Journal Le Monde du 28/9/87)

la Cour... avait choisi la première heure du trentième jour du procès. Dimanche à 1 heure du matin... M. Hachemi Zammel prononce le verdict...(Journal Le Monde du 29/9/87)

La cour s'était retirée pour délibérer jeudi en milieu de matinée...(Journal Le Monde du 29/9/87)

On attendait le verdict pour la nuit suivante...(Journal Le Monde du 29/9/87)

Il s'agit de M.Mohamed Treschi, trente-trois ans, condamné dimanche à vingt ans de travaux forcés.(Journal Le Monde du 29/9/87)

Le jour même -le jeudi 3 septembre- où... Mgr Evariste Ngoyagoye... informait le pape...(Journal Le Monde du 30/9/87)

les dirigeants iraniens... n'avaient cessé d'annoncer comme imminente «l'année de la victoire»...(Journal Le Monde du 4/6/88)

Huit années d'une interminable et sanglante guerre... sont venues à bout du moral d'une armée...(Journal Le Monde du 4/6/88)

au fil d'épopées dans lesquelles le siècle prochain plongera ses racines.(Journal Le Monde du 2/6/88)

1988 s'annonce, pour le Vietnam, comme l'une des années les plus difficiles depuis la victoire communiste de 1975...(Journal Le Monde du 3/6/88)

Tout cela pourrait permettre au Vietnam, dans les années qui viennent, d'amorcer son développement...(Journal Le Monde du 3/6/88)

Une France... du bébé obligatoire tous les deux ans, au pis tous les trente mois...(Journal Le Monde du 11/3/88)

Beaucoup de commandants... sont assis sur des arsenaux qui leur permettraient de tenir au-delà du délai de douze mois dont on parle...(Journal Le Monde du 28/12/87)

Des membres repentis de Prima Linea... risquent, demain, d'être réveillés en sursaut à l'heure légale...(Journal Le Monde du 5/9/87)
deux dissidents... sont morts, le premier sous la torture en juin 1986, le second exécuté en mars 1987.(Sélection hebdomadaire du journal Le Monde du 28/9/87)
 le ton du débat politique se durcit de jour en jour.(Journal Le Monde du 2/10/87)
 M. José Sarney... prévoit de porter la croissance de 5% cette année à 7% à compter de 1989...(Journal Le Monde du 5/10/87)
 un radiotélescope géant... pour identifier les molécules errant à des années-lumière de notre planète.(Journal Le Monde du 23/9/87)
 tout finit par se savoir et, de nos jours, de plus en plus vite et partout.(Journal Le Monde du 24/11/87)
 Initialement prévue du 7 au 9 décembre, cette conférence aura lieu du 14 au 16 décembre.(Journal Le Monde du 24/11/87)
 M. Gorbatchev a jugé «décisives les deux à trois prochaines années».(Journal Le Monde du 3/11/87)
 Staline est crédité d'avoir «fait triompher le léninisme» dans la seconde moitié des années 20...(Journal Le Monde du 4/11/88)
 Vous vous croiriez, certaines heures de pointe, sur quelque avenue parisienne...(Journal Le Monde du 31/10/87)
A la nuit tombée... l'ambiance est sinistre.(Journal Le Monde du 31/10/87)
 nous étions en octobre, à la belle saison. Qu'en sera-t-il au coeur de l'hiver?(Journal Le Monde du 31/10/87)
 même si par miracle ils étaient devenus sincères, ce n'est pas demain la veille qu'ils apprendraient à gouverner.(Journal Le Monde du 30/10/87)
Hier -il y a deux ans encore- tout était simple.(Journal Le Monde du 30/10/87)
 Tout bénéfice pour le général Jaruzelski? Sur quelques mois, un ou deux ans peut-être...(Journal Le Monde du 30/10/87)
 les espérances de 1980 ne sont plus à l'ordre du jour.(Journal Le Monde du 30/10/87)
 les résultats tangibles dans la vie de tous les jours sont minces.(Journal Le Monde du 31/10/87)
 Pas du jour au lendemain sans doute, car, estime Vitia, «on ne change pas si vite un système économique».(Journal Le Monde du 31/10/87)
 Afghanistan: l'après-15 mai(Journal Le Monde du 16/5/88)
 j'ai mis un an avant de réaliser que j'étais prisonnier...(Journal Le Monde du 13/5/88)
 à l'approche de l'élection présidentielle libanaise, prévue pour septembre, les extrémistes... font... figure de gêneurs...(Journal Le Monde du 17/5/88)
 peu après 1 heure du matin (7 heures, heure française), le vendredi 4 décembre, les mille cent cinq détenus... se sont rendus...(Journal Le Monde du 5/12/87)

| |
|-----------------------|
| INDEX DES PROGRAMMES. |
|-----------------------|

| | | | | | |
|---------------|----|---------------|----|---------------------|----|
| alpha..... | 31 | ec_car_e..... | 84 | maj_min..... | 32 |
| alphacct..... | 32 | ecralist..... | 62 | majus..... | 27 |
| car_int..... | 84 | ecran..... | 28 | majuscul..... | 84 |
| cas_lrch..... | 45 | ecrire..... | 27 | MAXCHAIN..... | 85 |
| cat_ch..... | 84 | ecrit..... | 84 | MAXCHIF..... | 85 |
| chargdic..... | 47 | efface_e..... | 29 | MAXDNUM..... | 85 |
| chargnom..... | 47 | entetcor..... | 83 | MAXETATS..... | 85 |
| CHOIX..... | 85 | est_lt..... | 84 | MAXFICH..... | 84 |
| chrchdic..... | 49 | est_ltnb..... | 27 | MAXGRAM..... | 85 |
| cl_mots..... | 6 | est_maj..... | 84 | MAXLIST..... | 85 |
| cla_choi..... | 55 | est_min..... | 84 | MAXMOT..... | 85 |
| clf_choi..... | 53 | est_minu..... | 32 | MAXPART..... | 85 |
| comp_ch..... | 84 | est_nb..... | 84 | MAXPONCT..... | 85 |
| compauto..... | 18 | ferme..... | 84 | MAXTEMPS..... | 85 |
| comptnoe..... | 57 | ferme_f..... | 84 | memoire..... | 84 |
| contr_ca..... | 60 | fermetur..... | 31 | menu..... | 38 |
| contr_dd..... | 66 | FICHIER..... | 84 | min_maj..... | 34 |
| contr_de..... | 73 | FIN..... | 84 | minuscule..... | 84 |
| contr_di..... | 82 | fin_adv..... | 43 | mot_comp..... | 50 |
| contr_do..... | 76 | fin_impr..... | 30 | MOTS..... | 85 |
| contr_dt..... | 68 | imprime..... | 29 | ouv_a_f..... | 84 |
| contr_er..... | 81 | INDEX..... | 85 | ouv_ec_f..... | 84 |
| contr_gp..... | 77 | ini_f..... | 84 | ouv_l_f..... | 84 |
| contr_le..... | 81 | ini_impr..... | 30 | ouvertur..... | 31 |
| contr_mo..... | 63 | ins_dico..... | 44 | ouvre_ec..... | 84 |
| contr_nt..... | 80 | l_ch..... | 84 | part_ch..... | 33 |
| contr_pd..... | 67 | lib_dico..... | 57 | passage..... | 58 |
| contrali..... | 62 | libauto..... | 55 | pos_debf..... | 84 |
| cop_ch..... | 84 | libere..... | 84 | posit_e..... | 28 |
| cp_choi..... | 41 | LIGNINDX..... | 85 | présenta..... | 85 |
| cp_dico..... | 40 | lire_ent..... | 58 | RIEN..... | 84 |
| cp_indx..... | 39 | lire_tab..... | 59 | romain(chaine)..... | 35 |
| cp_mots..... | 39 | lit_ca_f..... | 84 | tabtrans..... | 23 |
| cp_tran..... | 40 | lit_cl..... | 84 | taille..... | 84 |
| creeindx..... | 3 | lit_indx..... | 43 | termine..... | 84 |
| ct_tran..... | 52 | lit_mot..... | 42 | test_car..... | 33 |
| debut..... | 43 | lit_nbca..... | 43 | test_iem..... | 36 |
| DICO..... | 85 | lit_term..... | 41 | testauto..... | 51 |
| dnum..... | 12 | lit_veri..... | 37 | titre..... | 37 |
| droite..... | 56 | main..... | 1 | TRANS..... | 85 |

| |
|----------------------------|
| TABLE DES MATIERES. |
|----------------------------|

VOLUME I.

| | |
|--|-----------|
| 1. Introduction. | 1 |
| 1.1. Informatique et linguistique..... | 1 |
| 1.2. Dictionnaire et lexique-grammaire..... | 7 |
| 1.2.1. <i>Les dictionnaires du LADL.</i> | 7 |
| 1.2.2. <i>La constitution du lexique-grammaire.</i> | 7 |
| 1.2.3. <i>La théorie des automates finis.</i> | 8 |
| 1.3. Interprétation sémantique..... | 10 |
| 1.3.1. <i>De la syntaxe à la sémantique.</i> | 10 |
| 1.3.2. <i>La notion de temps.</i> | 10 |
| 2. Présentation du travail. | 14 |
| 2.1. Le dictionnaire..... | 14 |
| 2.1.1. <i>Les informations morphologiques et typographiques.</i> | 14 |
| 2.1.2. <i>Le passage de la lecture à la consultation.</i> | 15 |
| 2.1.3. <i>L'ordre des entrées.</i> | 17 |
| 2.1.4. <i>La structure d'un article du dictionnaire.</i> | 17 |
| 2.2. Le lexique-grammaire..... | 19 |
| 2.2.1. <i>Les informations syntaxiques.</i> | 19 |
| 2.2.3. <i>L'automate.</i> | 19 |
| 2.2.2. <i>Les tables de transitions.</i> | 19 |
| 2.3. Le programme de lecture..... | 22 |
| 2.3.1. <i>Le début du programme.</i> | 22 |
| 2.3.2. <i>Les conventions et les options du programmes.</i> | 23 |
| 2.3.3. <i>Le module de consultation du dictionnaire.</i> | 23 |
| 2.3.4. <i>Le module de consultation de l'automate.</i> | 28 |
| 2.3.5. <i>Le module de consultation des tables de transitions.</i> | 31 |
| 3. Dates et adverbess de temps. | 35 |
| 3.1. Adverbess de date..... | 35 |
| 3.1.1. <i>Jour, mois, année.</i> | 35 |
| 3.1.2. <i>Heure.</i> | 37 |
| 3.2.3. <i>Figures 6 à 14.</i> | 37 |
| 3.2. Adverbess de temps apparentés à une date..... | 43 |
| 3.2.1. <i>Adverbess de temps.</i> | 43 |
| 3.2.2. <i>Généralisation de la notion d'adverbe.</i> | 45 |
| 3.3. Verbes supports..... | 46 |
| 4. Compléments circonstanciels. | 48 |
| 4.1. Formes simples..... | 48 |
| 4.1.1. <i>Prépositions.</i> | 48 |
| 4.1.2. <i>Principaux déterminants.</i> | 48 |
| 4.1.3. <i>Noms de temps.</i> | 50 |
| 4.1.4. <i>Modifieurs.</i> | 51 |
| 4.1.5. <i>Noms de fêtes.</i> | 55 |

| | |
|--|------------|
| 4.2. Autres déterminants..... | 58 |
| 4.2.1. Déterminants adjectivaux..... | 58 |
| 4.2.2. Prédéterminants..... | 58 |
| 4.2.3. Déterminants adverbiaux..... | 59 |
| 4.2.4. Déterminants nominaux..... | 59 |
| 4.2.5. Localisation dans le temps..... | 60 |
| 4.2.6. Article indéfini un..... | 62 |
| 4.2.7. Noms de fête..... | 62 |
| 4.2.8. Déterminants et Prépositions..... | 62 |
| 4.2.9. Combinaisons de déterminants..... | 63 |
| 4.2.10. Automates..... | 64 |
| 4.3. Formes complexes..... | 64 |
| 4.3.1. Compléments de noms..... | 64 |
| 4.3.2. Prépositions et conjonctions..... | 67 |
| 4.3.3. Ellipse et pronominalisation..... | 68 |
| 5. Verbe support avoir lieu..... | 70 |
| 5.1. Adverbes de date..... | 70 |
| 5.2. Compléments circonstanciels: formes simples..... | 70 |
| 5.3. Déterminants adjectivaux..... | 73 |
| 5.4. Déterminants nominaux..... | 75 |
| 5.5. Localisation dans le temps..... | 80 |
| 5.6. Article indéfini un..... | 84 |
| 5.7. Noms de fête..... | 84 |
| 6. Conclusion..... | 85 |
| 6.1. Vers un début d'interprétation..... | 85 |
| 6.2. Exemple de maintenance..... | 87 |
| 6.3. Résultats..... | 87 |
| 6.4. Perspectives..... | 93 |
| 7. Annexe I: Automates..... | 94 |
| 7.1. Automate des dates..... | 94 |
| 7.2. Automate des Noms de temps et Adverbes de date..... | 96 |
| 7.3. Automate des Noms de fête..... | 97 |
| 7.4. Automate final..... | 98 |
| 8. Annexe II: Tables de transitions..... | 104 |
| 8.1. Prépositions..... | 104 |
| 8.1.1. <i>Prép Date</i> | 104 |
| 8.1.2. <i>Prép Advtps</i> | 105 |
| 8.1.3. <i>Prép Ntps</i> | 106 |
| 8.1.4. <i>Prép les Ntps</i> | 107 |
| 8.1.5. <i>Prép ces Ntps</i> | 108 |
| 8.1.6. <i>Prép les dnum Ntps</i> | 109 |
| 8.1.7. <i>Prép ces dnum Ntps</i> | 110 |
| 8.1.8. <i>Prép LE Ntps</i> | 111 |
| 8.1.9. <i>Prép CE Ntps</i> | 112 |
| 8.1.10. <i>Prép dnum Ntps</i> | 113 |
| 8.2. Modifieurs..... | 114 |
| 8.2.1. <i>Ntps Modif</i> | 114 |
| 8.2.2. <i>Modif Ntps</i> | 115 |
| 8.2.3. <i>Date Modif</i> | 116 |
| 8.2.4. <i>Modif Date</i> | 117 |

| | |
|--|-----|
| 8.3. Appositions..... | 118 |
| 8.3.1. <i>Advps Ntps</i> | 118 |
| 8.3.2. <i>Ntps Ntps</i> | 119 |
| 8.4. Noms de fête..... | 120 |
| 8.4.1. <i>Nfête Modif</i> | 120 |
| 8.4.2. <i>Modif Nfête</i> | 121 |
| 8.4.3. <i>Prép LE Nfête</i> | 122 |
| 8.4.4. <i>Prép CE Nfête</i> | 123 |
| 8.4.5. <i>Prép Nfête</i> | 124 |
| 8.5. Prédéterminants..... | 125 |
| 8.5.1. <i>Préd Ntps</i> | 125 |
| 8.5.2. <i>Ntps Préd</i> | 126 |
| 8.5.3. <i>Préd les Ntps</i> | 127 |
| 8.5.4. <i>Préd Ces Ntps</i> | 128 |
| 8.5.5. <i>Préd les dnum Ntps</i> | 129 |
| 8.5.6. <i>Préd ces dnum Ntps</i> | 130 |
| 8.5.7. <i>Préd LE Ntps</i> | 131 |
| 8.5.8. <i>Préd CE Ntps</i> | 132 |
| 8.5.9. <i>Préd des Ntps</i> | 133 |
| 8.5.10. <i>Préd dnum Ntps</i> | 134 |
| 8.5.11. <i>Préd Date</i> | 135 |
| 8.5.12. <i>Date Préd</i> | 136 |
| 8.6. Déterminants..... | 137 |
| 8.6.1. <i>Dét Ntps</i> | 137 |
| 8.6.2. <i>Dét de Ntps</i> | 138 |
| 8.6.3. <i>Dét de (les + des) Ntps</i> | 139 |
| 8.6.4. <i>Dét de ces Ntps</i> | 140 |
| 8.6.5. <i>Dét de les dnum Ntps</i> | 141 |
| 8.6.6. <i>Dét de ces dnum Ntps</i> | 142 |
| 8.6.7. <i>Dét de LE Ntps</i> | 143 |
| 8.6.8. <i>Dét de CE Ntps</i> | 144 |
| 8.6.9. <i>Dét de dnum Ntps</i> | 145 |
| 8.6.10. <i>Dét Date</i> | 146 |
| 8.6.11. <i>Dét de Date</i> | 147 |
| 8.6.12. <i>Dét de (LE + CE + E) Nfête</i> | 148 |
| 8.7. Prépositions et Déterminants..... | 149 |
| 8.7.1. <i>Prép (Préd + Dét)</i> | 149 |
| 8.7.2. <i>Prép Prép</i> | 150 |
| 8.7.3. <i>Prép Dét</i> | 151 |
| 8.7.4. <i>Prép Dét (pour les Noms de fête)</i> | 152 |
| 8.8. Combinaison de déterminants..... | 153 |
| 8.8.1. <i>Advd (Préd + Dét)</i> | 153 |
| 8.8.2. <i>Advd Dét</i> | 154 |
| 8.8.3. <i>(Préd + Dét)(Préd + Dét)</i> | 155 |
| 8.8.4. <i>Dét Dét</i> | 156 |
| 8.8.5. <i>Dét de Dét</i> | 157 |
| 8.9. Combinaison de déterminants (Noms de fête)..... | 158 |
| 8.9.1. <i>Advd Dét</i> | 158 |
| 8.9.2. <i>Dét Dét</i> | 159 |
| 8.9.3. <i>Dét de Dét</i> | 160 |
| 8.10. Compléments de noms..... | 161 |
| 8.10.1. <i>Ntps de Advps</i> | 161 |
| 8.10.2. <i>Ntps de Ntps</i> | 162 |
| 8.10.3. <i>Date de Ntps</i> | 163 |
| 8.10.4. <i>Ntps de Date</i> | 164 |
| 8.10.5. <i>Date de Date</i> | 165 |

| | |
|---|------------|
| 8.11. Compléments de noms (Noms de fête)..... | 166 |
| 8.11.1. <i>Lundi de Nfête</i> | 166 |
| 8.11.2. <i>Ntps de Nfête</i> | 167 |
| 8.11.3. <i>Nfête de Advtps</i> | 168 |
| 8.11.4. <i>Nfête de Ntps</i> | 169 |
| Index. | 170 |
| Bibliographie. | 171 |
| Table des matières | 177 |

VOLUME II

| | |
|--|------------|
| 9. Programmes. | 1 |
| 9.1. Fichier ADVDATE.C..... | 1 |
| 9.2. Fichier MODULE0.C..... | 3 |
| 9.3. Fichier MODULE1.C..... | 6 |
| 9.4. Fichier MODULE1A.C..... | 12 |
| 9.5. Fichier MODULE2.C..... | 18 |
| 9.6. Fichier MODULE3.C..... | 23 |
| 9.7. Fichier ASCIIDOS.C..... | 27 |
| 9.8. Fichier OUTILS.C..... | 31 |
| 9.9. Fichier BIBLI.H..... | 84 |
| 9.10. Fichier DEFINIR.H..... | 85 |
| 10. Fichiers Principaux. | 87 |
| 10.1. Fichier ADVDATE.NOM..... | 87 |
| 10.2. Fichier ADVDATE.IND..... | 87 |
| 10.3. Fichier ADVDATE.DIC..... | 87 |
| 10.4. Fichier ADVDATE.AUT..... | 98 |
| 11. Fichiers d'en-tête des tables de transitions. | 99 |
| 11.1. Fichier AD.ENT..... | 99 |
| 11.2. Fichier CA.ENT..... | 99 |
| 11.3. Fichier DT.ENT..... | 100 |
| 11.4. Fichier ER.ENT..... | 100 |
| 11.5. Fichier MO.ENT..... | 100 |
| 11.6. Fichier NF.ENT..... | 100 |
| 11.7. Fichier NT.ENT..... | 101 |
| 11.8. Fichier PD.ENT..... | 101 |
| 11.9. Fichier PR.ENT..... | 101 |
| 11.10. Fichier UN.ENT..... | 101 |
| 12. Fichiers corps des tables de transitions. | 102 |
| 12.1. Fichier ADNT.TAB..... | 102 |
| 12.2. Fichier AVDT.TAB..... | 102 |
| 12.3. Fichier AVPD.TAB..... | 102 |
| 12.4. Fichier CADECA.TAB..... | 102 |
| 12.5. Fichier CADENT.TAB..... | 102 |
| 12.6. Fichier CAMO.TAB..... | 103 |
| 12.7. Fichier CAPD.TAB..... | 104 |
| 12.8. Fichier DAER.TAB..... | 104 |
| 12.9. Fichier DTCA.TAB..... | 104 |

| | |
|-----------------------------------|-----|
| 12.10. Fichier DTDCDNNT.TAB | 105 |
| 12.11. Fichier DTDECA.TAB..... | 106 |
| 12.12. Fichier DTDECENF.TAB | 106 |
| 12.13. Fichier DTDECENT.TAB | 107 |
| 12.14. Fichier DTDECSNT.TAB..... | 108 |
| 12.15. Fichier DTDEDNNT.TAB | 109 |
| 12.16. Fichier DTDEDT.TAB..... | 110 |
| 12.17. Fichier DTDELENF.TAB | 110 |
| 12.18. Fichier DTDELENT.TAB | 111 |
| 12.19. Fichier DTDELSNT.TAB..... | 112 |
| 12.20. Fichier DTDENF.TAB | 112 |
| 12.21. Fichier DTDENT.TAB..... | 113 |
| 12.22. Fichier DTDLDNNT.TAB..... | 114 |
| 12.23. Fichier DTDSNT.TAB | 115 |
| 12.24. Fichier DTDT.TAB..... | 115 |
| 12.25. Fichier DTNT.TAB..... | 116 |
| 12.26. Fichier DTPD.TAB..... | 116 |
| 12.27. Fichier MOCA.TAB..... | 116 |
| 12.28. Fichier MONF.TAB..... | 117 |
| 12.29. Fichier MONT.TAB..... | 117 |
| 12.30. Fichier NFDA.TAB..... | 117 |
| 12.31. Fichier NFDB.TAB..... | 118 |
| 12.32. Fichier NFDEAD.TAB..... | 118 |
| 12.33. Fichier NFDEDA.TAB..... | 118 |
| 12.34. Fichier NFDEDB.TAB..... | 118 |
| 12.35. Fichier NFDENT.TAB | 118 |
| 12.36. Fichier NFDO.TAB..... | 118 |
| 12.37. Fichier NFMO.TAB..... | 118 |
| 12.38. Fichier NJDENF.TAB | 118 |
| 12.39. Fichier NJNT.TAB..... | 118 |
| 12.40. Fichier NTDA.TAB..... | 119 |
| 12.41. Fichier NTDB.TAB..... | 119 |
| 12.42. Fichier NTDEAD.TAB | 119 |
| 12.43. Fichier NTDECA.TAB..... | 120 |
| 12.44. Fichier NTDENF.TAB | 120 |
| 12.45. Fichier NTDENT.TAB..... | 121 |
| 12.46. Fichier NTDO.TAB | 121 |
| 12.47. Fichier NTER.TAB..... | 122 |
| 12.48. Fichier NTMO.TAB..... | 122 |
| 12.49. Fichier NTNT.TAB..... | 123 |
| 12.50. Fichier NTPD.TAB | 124 |
| 12.51. Fichier PDCA.TAB..... | 125 |
| 12.52. Fichier PDCENT.TAB | 125 |
| 12.53. Fichier PDCSDNNT.TAB..... | 126 |
| 12.54. Fichier PDCSNT.TAB | 126 |
| 12.55. Fichier PDDNNT.TAB..... | 127 |
| 12.56. Fichier PDDSNT.TAB..... | 127 |
| 12.57. Fichier PDDT.TAB..... | 128 |
| 12.58. Fichier PDLENT.TAB..... | 129 |
| 12.59. Fichier PDLSDNNT.TAB..... | 129 |
| 12.60. Fichier PDLSNT.TAB | 130 |
| 12.61. Fichier PDNT.TAB | 130 |
| 12.62. Fichier PDPD.TAB | 131 |
| 12.63. Fichier PDPR.TAB..... | 131 |
| 12.64. Fichier PRAD.TAB..... | 131 |
| 12.65. Fichier PRCA.TAB..... | 132 |
| 12.66. Fichier PRCENF.TAB..... | 132 |
| 12.67. Fichier PRCENT.TAB..... | 132 |

| | |
|-----------------------------------|-----|
| 12.68. Fichier PRCSNNT.TAB..... | 133 |
| 12.69. Fichier PRCSNT.TAB | 133 |
| 12.70. Fichier PRDNNT.TAB | 134 |
| 12.71. Fichier PRDSNT.TAB..... | 134 |
| 12.72. Fichier PRDT.TAB | 135 |
| 12.73. Fichier PRLNF.TAB..... | 135 |
| 12.74. Fichier PRLNT.TAB..... | 136 |
| 12.75. Fichier PRLSDNNT.TAB | 136 |
| 12.76. Fichier PRLSNT.TAB..... | 137 |
| 12.77. Fichier PRNF.TAB..... | 137 |
| 12.78. Fichier PRNT.TAB | 137 |
| 12.79. Fichier PRPD.TAB..... | 138 |
| 12.80. Fichier PRUN.TAB..... | 138 |
| 12.81. Fichier UNNT.TAB..... | 138 |

| | |
|----------------------|------------|
| 13. Exemples. | 139 |
|----------------------|------------|

| | |
|-----------------------------|------------|
| Index des programmes | 145 |
| Table des matières | 146 |



