



THESIS / THÈSE

MASTER EN SCIENCES INFORMATIQUES

Contribution à la réalisation d'un système de conception automatisée d'un appel d'offres

Demaret, Ch.; Maystadt, L.

Award date:
1986

Awarding institution:
Universite de Namur

[Link to publication](#)

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal ?

Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Facultés Universitaires Notre-Dame de la Paix, Namur

Institut d'Informatique

Année académique 1985-1986

Contribution à la réalisation d'un
système de conception automatisée
d'un appel d'offres

Tome 3

Ch. Demaret

L. Maystadt

Promoteur : Ph. van Bastelaer

Mémoire présenté en vue de l'obtention du grade
de Licencié et Maître en Informatique.

ANNEXE 12.1. : MODULE "GESTION DES FOURNISSEURS"

- 1) - nom :
menu_fourn
- structure :
procEDURE
- mode d'appel :
menu_fourn
- utilise :
efecran, message, crea_arb_f, curseur, lirechar,
ajout_fourn, list_fourn, modif_av, supp_fourn,
exit_fourn, tr_fourn
- paramètres internes :
car : char : un caractère lu à l'écran
okset : set of char : ensemble des caractères
admissibles
- fonction :
propose, d'une manière interactive, de choisir dans
un menu une opération concernant la gestion des
fournisseurs.
- 2) - nom :
tr_fourn
- structure :
procEDURE
- mode d'appel :
tr_fourn
- besoin :
ensemble
- utilise :
verif_fourn, efecran, message, lirefourn, curseur,
crea_arb_f, det_arb_f, chng_disk, lirechar, sc_find_x,
sc_find_y, write, ecrfourn, exit, sc_erase_to_eol,
erreur, gotoxy, concat, lirestring
- paramètres internes :
switch_bd : boolean : indique la fin du fichier
des fournisseurs
trans : boolean : indique qu'un transfert a été
effectué
ok_present : boolean : indique la présence de four-
nisseur
d_fourn : r8 : record fournisseur
num_rec_f : integer : numéro de record
x1,x2,y1,y2 : integer : coordonnées de curseur
ci,ci2 : integer : variables pour la B.D.A.
car : char : un caractère lu à l'écran
okset : set of char : ensemble de caractères
admissibles
name_disk : string : nom de la disquette de travail

- fonction :
sélectionne, d'une manière interactive, à partir du dictionnaire des fournisseurs les fournisseurs à retenir.

- 3) - nom :
ajout_fourn
- structure :
procédure
- mode d'appel :
ajout_fourn
- besoin :
fournex
- utilise :
efecran, message, iif, curseur, lirechar, ecrfourn, aj_arb_f
- paramètres internes :
ajout : boolean : indique qu'un ajout a été effectué
okset : set of char : ensemble de caractères admissibles
car : char : un caractère lu à l'écran
ci : integer : variable pour la B.D.A.
num_rec_f : integer : numéro de record d'un fournisseur
d_fourn : r8 : record fournisseur
- fonction :
permet, d'une manière interactive, d'ajouter un fournisseur dans la B.D.A. et dans l'arbre des fournisseurs.

- 4) - nom :
iif
- structure :
procédure
- mode d'appel :
iif (d_fourn)
- sortie :
d_fourn : r8 : record fournisseur
- utilise :
isf, iaf
- paramètres internes :
nom_soc : string : nom du fournisseur
adr_et_tel : r81 : adresse, téléphone,... du fournisseur
etat_fourn : array [1..11] of integer : états d'avancements
i : integer : variable de travail d'une boucle
- fonction :
active des procédures de lecture de données et remplit "d_fourn" avec ces données.

- 5) - nom :
isf
- structure :
procédure
- mode d'appel :
isf (nom_soc)
- besoin :
fournex, ensemble
- modifie :
fournex
- sortie :
nom_soc : string : nom du fournisseur
- utilise :
curseur, lirestring, f_deja_intro, bip, erreur, exit
- fonction :
lit à la console la donnée "nom_soc" et stocke cette donnée dans "nom_soc", si le fournisseur existe déjà un message d'erreur est envoyé, "fournex" est mis à true et on retourne au menu principal.
- 6) - nom :
iaf
- structure :
procédure
- mode d'appel :
iaf (adr_et_tel)
- besoin :
numset, ensemble
- sortie :
adr_et_tel : r81 : adresse et téléphone du fournisseur
- utilise :
curseur, lirestring
- paramètres internes :
num_rec_f : string : numéro de rue
nom_rue_f : string : nom de rue
code_post_f : string : code postal
local_f : string : nom de localité
pref : string : préfixe
suf : string : suffixe
telex : string : télex
- fonction :
lit les données numéro de rue, nom de rue, localité,... à la console et les stocke dans "adr_et_tel".
- 7) - nom :
list_fourn
- structure :
procédure
- mode d'appel :
list_fourn
- besoins :
rac, rac2

- utilise :
efecran, message, curseur, lirechar, bip, erreur,
crea_arb_f, openprinter, prlongueur, list_arb_f,
det_arb_f, closeprinter
 - paramètres internes :
okset1, okset2 : set of char : ensembles de caractères admissibles
car1, car2, choix : char : caractères lus à l'écran
critere : integer : numéro de critère
x, y : integer : coordonnées de curseur
 - fonction :
permet de demander, de manière interactive, d'afficher à l'écran ou d'imprimer les informations concernant les fournisseurs.
- 8) - nom :
modif_av
- structure :
procedure
 - mode d'appel :
modif_av
 - utilise :
verif_fourn, efecran, message, curseur, lirechar,
init_et_av, mte, mle, erreur, exit, bip
 - paramètres internes :
okset : set of char : ensemble de caractères admissibles
car : char : un caractère lu à l'écran
 - fonction :
permet, d'une manière interactive, de choisir une des 3 fonctions de modification des états d'avancements (et ce pour les fournisseurs sélectionnés).
- 9) - nom :
init_et_av
- structure :
procedure
 - mode d'appel :
init_et_av
 - utilise :
efecran, message, curseur, lirechar, init_etats,
exit
 - paramètres internes :
okset : set of char : ensemble des caractères admissibles
car : char : un caractère lu à l'écran
 - fonction :
permet, de manière interactive, de demander s'il faut réinitialiser tous les états d'avancements des fournisseurs sélectionnés, si la réponse est positive alors activation de l'initialisation.

10) - nom :
m1e
- structure :
procédure
- mode d'appel :
m1e
- besoin :
ensemble
- utilise :
efecran, message, curseur, lirestring, f_deja_intro,
lirechar, parc_arb_f, lirenum, meaf, ecrnum, exit,
bip, erreur
- paramètres internes :
okset1, okset2 : set of char : ensemble de caractères
admissibles
car1, car2 : char : caractères lus à
l'écran
nom_société : string : nom du fournisseur
d_fourn : r8 : record fournisseur
num_rec_f : integer : numéro de record
- fonction :
gère la demande de modification d'un état d'avancement.

11) - nom :
mte
- structure :
procédure
- mode d'appel :
mte
- utilise :
efecran, message, curseur, lirefourn, meaf, lirechar,
modfourn
- paramètres internes :
okset : set of char : ensemble des caractères
admissibles
car : char : un caractère lu à
l'écran
switch_bd2 : boolean : indique fin du fichier
num_rec_f : integer : numéro de record
ci : integer : variable pour la B.D.A.
d_fourn : r8 : record fournisseur
- fonction :
gère la demande de modification de tous les états
d'avancements.

12) - nom :
meaf
- structure :
procédure
- mode d'appel :
meaf (d_fourn)
- sortie :
d_fourn : r8 : record fournisseur (contient l'état
d'avancement modifié)
- utilise :
efecran, message, curseur, write, lirechar, exit,
gotoxy
- paramètres internes :
okset : set of char : ensemble de caractères
admissibles
car : char : un caractère lu à l'écran
i, j : integer : variable de travail
x, y : integer : coordonnées de curseur
- fonction :
affiche les états d'avancements, lit et stocke
les modifications éventuelles venant de l'utilisateur.

12b) - nom :
exit_fourn
- structure :
procédure
- mode d'appel :
exit_fourn
- utilise :
det_arb_f
- fonction :
détruit l'arbre des fournisseurs.

13) - nom :
supp_fourn
- structure :
procédure
- mode d'appel :
supp_fourn
- besoins :
rac, ensemble
- modifie :
rac
- utilise :
efecran, message, curseur, lirestring, f_deja_intro,
verif_fourn, lirechar, lirenum, cernum, sup_arb_f,
bip, erreur

- paramètres internes :
 - okset1, okset2 : set of char : ensembles de caractères admissibles
 - car1, car2 : char : caractères lus à l'écran
 - nom_f : string : nom d'un fournisseur
 - num_rec_f : integer : numéro de record d'un fournisseur
 - d_fourn : r8 : record fournisseur
- fonction :
 - permet, de manière interactive, de supprimer un record fournisseur de la BD et de l'arbre des fournisseurs.

ANNEXE 12.2. : MODULE "INTRODUCTION DES DONNEES A.O."

- 14) - nom :
 intro_ao
- structure :
 procedure
- mode d'appel :
 intro_ao
- utilise :
 welc_ao
- fonction :
 gère, au plus haut niveau le scénario de l'introduction des données de la lettre d'intention.
- 15) - nom :
 welc_ao
- structure :
 procedure
- mode d'appel :
 welc_ao
- besoin :
 ndisk
- modifie :
 ndisk
- utilise :
 efecran, message, time, curseur, lirechar, lirestring, sc_find_x, sc_find_y, liredate, escrit, lirereturn, verif_j_b, transf_j_bord, chng_disk, init, verif_name, exit
- paramètres internes :
 okset : set of char : ensemble de caractères admissibles
 car : char : un caractère lu à l'écran
 nom_cli : string : nom de la disquette de travail
 date : string : date de création A.O.
 ok : boolean : true si nom disquette correspond
 x, y : integer : coordonnées de curseur
- fonction :
 gère les opérations de départ i.e. si c'est un nouveau A.O. modifier le nom de la disquette sinon vérifier si le nom correspond,..., de plus, demande et stocke la date de création de l'A.O.

16) - nom :
 idao0
 - structure :
 procedure
 - mode d'appel :
 idao0
 - besoin :
 routage
 - modifie :
 routage
 - utilise :
 consult_journal, idao1, idao2, maj_journal
 - fonction :
 "racine de l'arbre des procédures" niveau 0 .

17) - nom :
 idao1
 - structure :
 procedure
 - mode d'appel :
 idao1
 - besoin :
 routage
 - modifie :
 routage
 - utilise :
 consult_journal, idao4, idao5, maj_journal
 - fonction :
 "noeud de l'arbre des procédures" niveau 1 .

18) - nom :
 idao2
 - structure :
 procedure
 - mode d'appel :
 idao2
 - besoin :
 routage
 - modifie :
 routage
 - utilise :
 consult_journal, idao6, idao7, maj_journal
 - fonction :
 "noeud de l'arbre des procédures" niveau 1 .

19) - nom :
 idao4
 - structure :
 procedure
 - mode d'appel :
 idao4
 - besoin :
 routage
 - modifie :
 routage
 - utilise :
 consult_journal, iie, irq, iasj, maj_journal
 - fonction :
 "noeud de l'arbre des procédures" niveau 2 .

20) - nom :
 idao5
 - structure :
 procedure
 - mode d'appel :
 idao5
 - besoin :
 routage
 - modifie :
 routage
 - utilise :
 consult_journal, idao, id, maj_journal
 - fonction :
 "noeud de l'arbre des procédures" niveau 2 .

21) - nom :
 idao6
 - structure :
 procédure
 - mode d'appel :
 idao6
 - besoin :
 routage
 - modifie :
 routage
 - utilise :
 consult_journal, isc, iif, maj_journal
 - fonction :
 "noeud de l'arbre des procédures" niveau 2 .

- 22) - nom :
 idao7
 - structure :
 procedure
 - mode d'appel :
 idao7
 - besoin :
 routage
 - modifie :
 routage
 - utilise :
 consult_journal, iis, iijs, ige, maj_journal
 - fonction :
 "noeud de l'arbre des procédures" niveau 2 .
- 23) - nom :
 iic
 - structure :
 procedure
 - mode d'appel :
 iic
 - utilise :
 consult_journal, efecran, message, inc, iae, ite,
 inr, maj_journal, pres_bl
 - fonction :
 "noeud de l'arbre des procédures" niveau 3 .
- 24) - nom :
 irq
 - structure :
 procedure
 - mode d'appel :
 irq
 - besoin :
 routage
 - modifie :
 routage
 - utilise :
 consult_journal, irc, irq1, irq2, irq3, maj_journal,
 pres_bl
 - fonction :
 "noeud de l'arbre des procédures" niveau 3 .

25) - nom :
iasp
- structure :
procédure
- mode d'appel :
iasp
- besoin :
routage
- modifiée :
routage
- utilise :
consult_journal, ia, ialt, ids, iaa, iaq, maj_journal,
pres_bl
- fonction :
"noeud de l'arbre des procédures" niveau 3 .

26) - nom :
id
- structure :
procédure
- mode d'appel :
id
- besoin :
routage
- modifiée :
routage
- utilise :
consult_journal, iri, ipl, maj_journal, pres_bl
- fonction :
"noeud de l'arbre des procédures" niveau 3 .

27) - nom :
iip
- structure :
procédure
- mode d'appel :
iip
- besoin :
routage
- modifiée :
routage
- utilise :
consult_journal, idu, ide, maj_journal, pres_bl
- fonction :
"noeud de l'arbre des procédures" niveau 3 .

- 28) - nom :
isc
- structure :
procédure
- mode d'appel :
isc
- besoin :
routage
- modifiée :
routage
- utilise :
consult_journal, pres_bl, ip, idc, iae, maj_journal
- fonction :
"noeud de l'arbre des procédures" niveau 3 .
- 29) - nom :
idaa
- structure :
procédure
- mode d'appel :
idaa
- besoin :
routage
- modifiée :
routage
- utilise :
consult_journal, pres_bl, idac, idr, maj_journal
- fonction :
"noeud de l'arbre des procédures" niveau 3 .
- 30) - nom :
iis
- structure :
procédure
- mode d'appel :
iis
- besoin :
routage
- modifiée :
routage
- utilise :
consult_journal, iig, itr, ibed, iio, maj_journal,
pres_bl
- fonction :
"noeud de l'arbre des procédures" niveau 3 .

31) - nom :
 iipe
 - structure :
 procédure
 - mode d'appel :
 iipe
 - besoin :
 routage
 - modifiée :
 routage
 - utilise :
 consult_journal, pres_bl, iie, iii, iap, iex, idp,
 maj_journal
 - fonction :
 "noeud de l'arbre des procédures" niveau 3 .

32) - nom :
 iqc
 - structure :
 procédure
 - mode d'appel :
 iqc
 - besoin :
 routage
 - modifiée :
 routage
 - utilise :
 consult_journal, pres_bl, iqa, maj_journal
 - fonction :
 "noeud de l'arbre des procédures" niveau 3 .

33) - nom :
 pres_bl
 - structure :
 procédure
 - mode d'appel :
 pres_bl (I)
 - entrée :
 I : integer : numéro de section
 - utilise :
 efecran, message, gotoxy, write, lirereturn
 - fonction :
 affiche un écran introductif par section.

34) - nom :
i_applic
- structure :
procédure
- mode d'appel :
i_applic
- besoins :
enscn, cr
- utilise :
efecran, message, curseur, sc_find_x, sc_find_y,
texte_libre, ecritt1
- paramètres internes :
ens_texte : set of char : ensemble de caractères
admissibles
x, y : integer : coordonnées de curseur
y2 : integer : numéro de ligne
bid : integer : variable bidon
ct : boolean : variable pour la BD
descrip : string : description des applica-
tions
- fonction :
lecture et stockage de la description des applica-
tions pour la L.I.

35) - nom :
verif_j_b
- structure :
procédure
- mode d'appel :
verif_j_b
- utilise :
reset, ioresult, message, lirereturn, close
- paramètres internes :
fich_journ : fu : fichier journal de bord
- fonction :
vérifie l'existence du journal de bord sur la dis-
quette de travail, s'il existe un message est
affiché.

36) - nom :
liredate
- structure :
procédure
- mode d'appel :
liredate (x, y, date)
- entrée :
x, y : integer : coordonnées de curseur
- sortie :
date : string : date lue

- utilise :
gotoxy, write, lireentier, erreur, str, insert
- paramètres internes :
jj, mm, aa : integer : jour, mois, année
j, m, a : string : jour, mois, année
- fonction :
lecture aux coordonnées "(x, y)" d'une date "date".

37) - nom :
inc

- structure :
procedure
- mode d'appel :
inc
- besoin :
ensemble
- utilise :
curseur, lirestring, ecrit
- paramètres internes :
nom_cli : string : nom du client
bid : integer : variable bidon
- fonction :
lecture et stockage du nom du client.

38) - nom :
iac

- structure :
procedure
- mode d'appel :
iac
- besoins :
ensemble, alphaset, numset
- utilise :
curseur, lirestring, écrit
- paramètres internes :
num_rue : string : numéro de rue
nom_rue : string : nom de rue
code_post : string : code postal
nom_local : string : nom de localité
bid : integer : variable bidon
- fonction :
lecture et stockage de l'adresse du client.

- 39) - nom :
itc
- structure :
procédure
- mode d'appel :
itc
- besoin :
numset
- utilise :
curseur, lirestring, insert, écrit
- paramètres internes :
pref, suf : string : préfixe et suffixe du numéro
de téléphone
tel : string : numéro de téléphone
bid : integer : variable bidon
- fonction :
lecture et stockage du numéro de téléphone du
client.
- 40) - nom :
inr
- structure :
procédure
- mode d'appel :
inr
- besoin :
alphaset
- utilise :
curseur, lirestring, lirechar, écrit
- paramètres internes :
okset : set of char : ensemble des caractères
admissibles
car : char : un caractère lu à l'écran
nom_resp : string : nom du responsable
titre_resp : string : titre du responsable
bid : integer : variable bidon
- fonction :
lecture et stockage du nom du responsable, ainsi
que de son titre.
- 41) - nom :
irc
- structure :
procédure
- mode d'appel :
irc
- besoins :
ensemble, numset
- utilise :
consult_journal, efecran, message, curseur,
lirechar, lirestring, écrit, maj_journal, susp_ao

- paramètres internes :
 - okset : set of char : ensemble de caractères admissibles
 - car, car2: char : caractères lus à l'écran
 - anc_sys : string : nom ancien système
 - an_inst : string : année installation ancien système
 - conf_inf : boolean : existence ancienne configuration
 - dem_mat : boolean : démontage matériel
 - bid : integer : variable bidon
- fonction :
lecture et stockage des variables de décision concernant un ancien système et du nom et de l'année d'installation de cet ancien système (s'il existe).

- 42) - nom :
irq1
- structure :
procédure
 - mode d'appel :
irq1
 - utilise :
consult_journal, efecran, message, curseur, lirechar, écrit, maj_journal, susp_ao
 - paramètres internes :
 - okset : set of char : ensemble de caractères admissibles
 - car1, car2, car3, car4 : char : caractères lus à l'écran
 - form_techn : boolean : formation technique du personnel
 - mat_simil : boolean : matériel similaire
 - amen_loc : boolean : aménagement des locaux
 - moy_com : boolean : installation des moyens de communication
 - bid : integer : variable bidon
 - fonction :
lecture et stockage des variables de décision décrites ci-dessus.

- 43) - nom :
irq2
- structure :
procédure
 - mode d'appel :
irq2

- utilise :
consult_journal, efecran, message, curseur,
lirechar, escrit, maj_journal, susp_ao
- paramètres internes :
 - okset : set of char : ensemble de caractères
admissibles
 - car1, car2 : char : caractères lus à l'écran
 - act_simul : boolean : activités simultanées
possible
 - prot_sys : boolean : protection du système
désirée
 - bid : integer : variable bidon
- fonction :
lecture et stockage des variables de décision
décrites ci-dessus.

- 44) - nom :
irq3
- structure :
procédure
 - mode d'appel :
irq3
 - utilise :
consult_journal, efecran, message, curseur, lirechar,
escrit, maj_journal, susp_ao, liref24
 - paramètres internes :
 - okset : set of char : ensemble de caractères
admissibles
 - car1, car2 : char : caractères lus à l'écran
 - conf_inf : boolean : existence d'une configura-
tion informatique
 - dev_prop : boolean : développements propres
désirés
 - rec_op : boolean : reconversion désirée
 - bid : integer : variable bidon
 - fonction :
lecture et stockage des variables décision "dev_prop"
et "rec_op" (si existence d'une configuration infor-
matique).

- 45) - nom :
ia
- structure :
procédure
 - mode d'appel :
ia
 - besoins :
enscar, cr


```
- utilise :
consult_journal, efecran, message, curseur, calcster,
lirestring, escrit, sc_find_x, sc_find_y, mod,
sc_cras_eos, gotoxy, write, maj_journal, susp_ao
- paramètres internes :
ens_texte, ens_texte2 : set of char : ensemble de
                        caractères admissibles
x, y                   : integer : coordonnées de
                        curseur
z                       : integer : numéro de ligne
nbr_ligne              : integer : nombre de lignes
                        (compteur)
num_serv              : integer : numéro de service
did                   : integer : variable bidon
nom_serv              : string  : nom de service
nom_appl              : string  : nom d'application
ster                  : string  : "CR"
- fonction :
lecture et stockage du numéro et du nom des services,
ainsi que du nom des applications relatives à cha-
que service.
- algorithmes :
  si consult_journal (23) = 0
  alors - ens_texte := enscar; calcster (ster);
         répéter
         - nettoyer l'écran et afficher un message;
         - positionner le curseur;
         - lire un string := nom_serv;
         - si not (cr in ens_texte) alors ens_texte :=
           ens_texte + cr finsi
         - si nom_serv ≠ ster
           alors escrit (2584, nom_serv, 0, false, bid,
             num_serv) finsi;
           si nom_serv ≠ ster
           alors - positionner le curseur;
                 - x := sc_find_x; y := sc_find_y;
                 - z := y, ens_texte2 := enscar;
                 nbr_ligne := y;
                 répéter
                 - si nbr_ligne mod 22 = 0
                   alors - effacer le bas de l'écran
                           à partir y;
                           - z := y
                   finsi
                 - positionner le curseur en (x-2,z);
                 - écrire un "-$";
                   (x seule hypothèse à connaître x)
                 - positionner le curseur en (x,z);
                 - lire un string := nom_appl;
                 - z := z + 1;
                 - nbr_ligne := nbr_ligne + 1;
```



```

- si not (cr in ens_texte2)
  alors ens_texte2 := ens_texte2 + cr
  finsi;
- si nom_appl ≠ ster
  alors écrire dans la B.D.A.
  finsi
  jusqu'à nom_appl = ster
finsi;
- jusqu'à nom_serv = ster
- maj_journal (23);
- susp_ao
finsi

```

Commentaires :

Le fait de retenir la position initiale du curseur et de la faire évoluer a été réalisé parce qu'il était impossible de retenir toutes les positions des lignes dans une matrice de coordonnées; la seule chose à connaître du fichier des écrans est dans ce cas : "-\$" au début de chaque ligne.

Ce principe est applicable pour les textes libres.

- 46) - nom :
- ialt
 - structure :
 - procedure
 - mode d'appel :
 - ialt
 - besoins :
 - ensemble, cr
 - utilise :
 - consult_journal, efecran, message, curseur, lirestring, sc_find_x, sc_find_y, écrit, gotoxy, write, maj_journal, susp_ao, calcster
 - paramètres internes :
 - ens_texte : set of char. : ensemble de caractères admissibles
 - x, y : integer : coordonnées de curseur
 - z : integer : numéro de ligne
 - nbr_ligne : integer : nombre de lignes (compteur)
 - bid : integer : variable bidon
 - ct1 : integer : variable pour la BD
 - nom_appl : string : nom d'application
 - ster : string : = "CR"
 - fonction :
 - lecture et stockage du nom des applications à long terme.

47) - nom :
ids
- structure :
procedure
- mode d'appel :
ids
- besoins :
enscar, cr
- utilise :
consult_journal, efecran, message, curseur, liref25,
sc_find_x, sc_find_y, sc_erase_to_eol, write, texte_
libre, ecrloca, maj_journal, susp_ao
- paramètres internes :
ens_texte : set of char : ensemble de caractères
admissibles

x, y : integer : coordonnées de curseur
y2 : integer : numéro de ligne
ci : integer : variable pour la BD
ct : boolean : variable pour la BD
fin : boolean : = true quand plus de service
des_serv : string : description d'un service
serv : r17 : record d'un service
- fonction :
lecture et stockage pour chaque service de sa
description.

48) - nom :
ip
- structure :
procedure
- mode d'appel :
ip
- besoins :
numset, enscar, cr
- utilise :
consult_journal, efecran, message, curseur,
lirestring, calcster, escrit, liref24, lirereturn,
sc_find_x, sc_find_y, ecritt1, sc_eras_eos,
gotoxy, write, maj_journal, susp_ao
- paramètres internes :
ens_texte : set of char : ensemble de caractères
admissibles

x, y : integer : coordonnées de curseur
z : integer : numéro de ligne
nbr_ligne : integer : nombre de lignes (compteur)
bid : integer : variable bidon
ct : boolean : variable pour la BD
form_techn : boolean : formation technique du
personnel
prest : string : prestation
nbr_per : string : nombre de personnes à
former

nbr_doc : string : nombre d'exemplaires de documents
ster : string : = "CR"
- fonction :
lecture et stockage des prestations, du nombre de personnes à former (si form_techn = true) et du nombre d'exemplaires des documents.

49) - nom :
ipl
- structure :
procedure
- mode d'appel :
ipl
- besoin :
enscar
- utilise :
consult_journal, efecran, message, curseur, liredate, sc_find_x, sc_find_y, escrit, lirechar, sc_eras_eos, gotoxy, write, lirestring, maj_journal, susp_ao
- paramètres internes :
okset : set of char : ensemble de caractères admissibles
i : integer : variable de travail
x, y : integer : coordonnées de curseur
z : integer : numéro de ligne
x2 : integer : numéro de colonne
nbr_ligne : integer : nombre de lignes (compteur)
bid : integer : variable bidon
car : char : un caractère lu à l'écran
ct : boolean : variable pour la BD
descript : string : description d'une date
date : string : date
- fonction :
lecture et stockage des dates du planning ainsi que des dates (et leur description) supplémentaires.

50) - nom :
idu
- structure :
procedure
- mode d'appel :
idu
- besoin :
ensemble
- utilise :
consult_journal, efecran, message, curseur, lirestring, escrit, maj_journal, susp_ao

- paramètres internes :
 - dur_gar_mat : string : durée de garantie de matériel
 - dur_gar_prest: string : durée de garantie de prestation
 - dur_gar_log : string : durée de garantie de logiciels
 - dur_conf : string : durée de confidentialité
 - dur_ass : string : durée d'assistance au démarrage
 - bid : integer : variable bidon
- fonction :
lecture et stockage des durées de garantie (matériel, prestation, logiciel), ainsi que des durées de confidentialité et d'assistance.

- 51) - nom :
ide
- structure :
procédure
 - mode d'appel :
ide
 - besoins :
ensemble, numset
 - utilise :
consult_journal, efecran, message, curseur, lirestring, ecrit, maj_journal, susp_ao
 - paramètres internes :
 - del_aband : string : délai d'abandon
 - del_ass_mat : string : délai essai matériel
 - del_ass_log : string : délai essai logiciel
 - del_arr_techn : string : délai arrivée technicien
 - del_engag : string : délai d'engagement
 - montant : string : montant d'amende
 - taux_dim : string : taux de diminution
 - bid : string : variable bidon
 - fonction :
lecture et stockage des délais d'abandon, d'essais (matériel et logiciel), d'engagement, d'arrivée du technicien, ainsi que du montant d'amende et du taux de diminution.

- 52) - nom :
iig
- structure :
procédure
 - mode d'appel :
iig

- besoins :
aphaset, ensemble, cr
- utilise :
consult_journal, efecran, message, curseur, calcster,
lirestring, escrit, maj_journal, susp_ao
- paramètres internes :
langue : string : langue admise
esp_vie : string : espérance de vie
mttr : string : mttr
mtbf : string : mtbf
temp_rep: string : augmentation temps de réponse
(batch)
bid : integer : variable bidon
ster : string : = "CR"
- fonction :
lecture et stockage de la langue admise, de l'espé-
rance de vie, du mttr, du mtbf et de l'augmentation
du temps de réponse (batch).

- 53) - nom :
idc
- structure :
procedure
 - mode d'appel :
idc
 - besoins :
cr, enskar
 - utilise :
consult_journal, efecran, message, liref24, lirechar,
sc_find_x, sc_find_y, sc_eras_eos, gotoxy, write,
lirestring, ecritt1, maj_journal, susp_ao, calcster
 - paramètres internes :
okset, ens_texte : set of char : ensemble de caractères admissibles
car : char : un caractère lu à l'écran
vis_ref : boolean : désir de visite de référence
reut_conf : boolean : réutilisation confi-
guration
ct : boolean : variable pour la BD
x, y : integer : coordonnées de curseur
z : integer : numéro de ligne
nbr_ligne : integer : nombre de lignes
(compteur)
bid : integer : variable bidon
crit : string : critère
ster : string : = "CR"
 - fonction :
affichage des critères, ensuite lecture et stockage
de nouveaux critères.

54) - nom :
 iaq
- structure :
 procedure
- mode d'appel :
 iaq
- besoins :
 enscar, cr
- utilise :
 consult_journal, efecran, message, curseur, sc_find_x,
 sc_find_y, texte_libre, ecritt1, maj_journal, susp_ao
- paramètres internes :
 ens_texte : set of char : ensemble de caractères
 admissibles
 x, y : integer : coordonnées de curseur
 y2 : integer : numéro de ligne
 bid : integer : variable bidon
 ct : boolean : variable pour la BD
 descrip_act : string : description activité
 entreprise
- fonction :
 lecture et stockage de la description de l'activité
 de l'entreprise.

55) - nom :
 iaa
- structure :
 procedure
- mode d'appel :
 iaa
- besoins :
 enscar, cr
- utilise :
 consult_journal, efecran, message, curseur, sc_find_x,
 sc_find_y, sc_erase_to_eal, liref17, write, texte_
 libre, ecrdescrip, liref11b, ecrf11b, maj_journal,
 susp_ao
- paramètres internes :
 ens_texte : set of char : ensemble de caractères
 admissibles
 x, y, x3, y3 : integer : coordonnées de curseur
 y2 : integer : numéro de ligne
 l, ci : integer : variables pour la BD
 ct, fin : boolean : variables pour la BD
 nom_app : string : nom d'application
 descrip_apl : string : description d'application
 nom_applic : r17 : record application
- fonction :
 lecture et stockage des descriptions des applications
 à court et long terme.

56) - nom :
iil
- structure :
procédure
- mode d'appel :
iil
- besoin :
ensemble
- utilise :
consult_journal, efecran, message, curseur, liref24,
lirestring, lirechar, escrit, maj_journal, susp_ao
- paramètres internes :
okset : set of char : ensemble de caractères
admissibles
car : char : un caractère lu à l'écran
ci : integer : variable pour la BD
bid : integer : variable bidon
langage : string : langage désiré
norme : string : norme du langage
composante : string : composantes du langage
dev_prop : boolean : désir de développements
propres
- fonction :
lecture et stockage du nom du langage, de la norme
et des composantes de ce langage si la variable
de décision "dev_prop" est vraie.

57) - nom :
idr
- structure :
procédure
- mode d'appel :
idr
- besoins :
ensemble, cr
- utilise :
consult_journal, efecran, message, curseur,
sc_find_x, sc_find_y, texte_libre, escrittl,
maj_journal, susp_ao, liref24.
- paramètres internes :
ens_texte : set of char : ensemble de caractères
admissibles
x, y : integer : coordonnées de curseur
y2 : integer : numéro de ligne
ct : boolean : variable pour la BD
descript : string : description de la recon-
version
conf_inf : boolean : existence d'une configu-
ration informatique
rec_ap : boolean : désir de reconversion
- fonction :
lecture et stockage de la description de la recon-
version s'il existe une configuration informatique
et s'il existe un désir de reconversion.

- 58) - nom :
itr
- structure :
procédure
- mode d'appel :
itr
- besoin :
ensemble
- utilise :
consult_journal, efecran, message, curseur, écrit,
lirestring, maj_journal, susp_ao
- paramètres internes :
- | | | |
|----------|---------------|---|
| okset | : set of char | : ensemble de caractères
admissibles |
| temp_rep | : string | : temps de réponse |
| imp_t_r | : string | : degré d'importance du temps
de réponse |
| bid | : integer | : variable bidon |
- fonction :
lecture et stockage du degré d'importance du temps
de réponse et du temps de réponse.
-
- 59) - nom :
ibed
- structure :
procédure
- mode d'appel :
ibed
- besoins :
ensemble, cr
- utilise :
consult_journal, efecran, message, curseur,
lirestring, écrit, maj_journal, susp_ao
- paramètres internes :
- | | | |
|-----------|---------------|---|
| ens_texte | : set of char | : ensemble de caractères
admissibles |
| bid | : integer | : variable bidon |
| back | : string | : temps de backup |
| act_entr | : string | : activités d'entretien
(période) |
| disp_nor | : string | : disponibilité normale |
| disp_exp | : string | : disponibilité exception-
nelle |
| disp_pour | : string | : nom pour lequel disponi-
bilité |
- fonction :
lecture et stockage du temps de backup, de la période
des activités d'entretien, de la disponibilité nor-
male et exceptionnelle (et nom pour lequel disponi-
bilité exceptionnelle).

60) - nom :
iex
- structure :
procédure
- mode d'appel :
iex
- besoins :
ensemble, cr, enscar
- utilise :
consult_journal, efecran, message, curseur,
lirestring, escrit, sc_find_x, sc_find_y, texte_
libre, maj_journal, susp_ao, calcster
- paramètres internes :
ens_texte, ens_texte2 : set of char : ensemble
de caractères admissibles
x, y : integer : coordonnées de
curseur
ster : string : = "CR"
y2 : integer : numéro de ligne
bid : integer : variable bidon
ct2 : integer : variable pour la BD
type_ext : string : type extension
ext_max : string : extension maximale
descrip : string : description extension
- fonction :
lecture et stockage du type d'extension, de l'exten-
sion maximale et de la description.

61) - nom :
iio
- structure :
procédure
- mode d'appel :
iio
- utilise :
consult_journal, efecran, message, curseur,
lirestring, escrit, susp_ao, maj_journal
- paramètres internes :
okset : set of char : ensemble de
caractères admissibles
rep1, rep2, rep3, rep4 : char : caractères lus à
l'écran
bid : integer : variable bidon
- fonction :
lecture et stockage de degrés d'importance concer-
nant le système.

62) - nom :
 iie
- structure :
 procedure
- mode d'appel :
 iie
- utilise :
 consult_journal, efecran, message, curseur, lirechar,
 ecrit, maj_journal
- paramètres internes :
 okset : set of char : ensemble de caractères
 admissibles
 rep1, rep2, rep3, rep4, rep5 : char : caractères
 lus à l'écran
 carac : boolean : type de caractères (maj,
 min)
 type_clav : boolean : type de clavier
 clavier : boolean : clavier (numérique,...)
 term_bloc : boolean : terminal en un bloc
 term_agr : boolean : terminal agréable
 bid : integer : variable bidon
- fonction
 lecture et stockage de réponses à des questions
 concernant le terminal.

63) - nom :
 iii
- structure :
 procedure
- mode d'appel :
 iii
- utilise :
 consult_journal, efecran, message, curseur,
 lirechar, écrit, maj_journal
- paramètres internes :
 okset : set of char : ensemble de caractères
 admissibles
 rep1, rep2, rep3, rep4, rep5, rep6, rep7 : char :
 caractères lus à l'écran
 carac : boolean : caractères à l'imprimante
 arrêt_a : boolean : arrêt automatique
 arrêt : boolean : arrêt en cas d'erreurs
 alim : boolean : alimentation en papier
 ex : boolean : plusieurs exemplaires
 chgt_pap : boolean : changement papier
 bruit : boolean : bruit
 bid : integer : variable bidon
- fonction :
 lecture et stockage de réponses à des questions
 concernant l'imprimante.

64) - nom :
iap
- structure :
procédure
- mode d'appel :
iap
- besoins :
numset, ensemble, cr
- utilise :
consult_journal, efecran, message, curseur,
lirestring, écrit, lirechar, sc_find_x, sc_find_y,
texte_libre, maj_journal, susp_ao
- paramètres internes :
okset, ens_texte : set of char : ensemble de caractères admissibles
car : char : un caractère lu à l'écran
ct : boolean : variable pour la BD
nbre_écr : string : nombre d'écrans
nbre_imp : string : nombre d'imprimantes
nom_per : string : nom de périphérique
nbre_per : string : nombre de périphériques
descrip : string : description de périphérique
x, y : integer : coordonnées de curseur
y2 : integer : numéro de ligne
bid : integer : variable bidon
- fonction :
lecture et stockage du nombre d'écrans et de périphériques, du nom, du nombre et de la description de périphériques.

65) - nom :
idp
- structure :
procédure
- mode d'appel :
idp
- besoins :
ensemble, cr
- utilise :
consult_journal, efecran, message, curseur, texte_libre, sc_find_x, sc_find_y, écrit, maj_journal, susp_ao

- paramètres internes :
 - ens_texte : set of char : ensemble de caractères admissibles
 - x, y : integer : coordonnées de curseur
 - y2 : integer : numéro de ligne
 - bid : integer : variable bidon
 - descrip : string : description des protections
 - ct : boolean : variable pour la BD
- fonction :
lecture et stockage de la description des protections.

- 66) - nom :
iri
- structure :
procédure
 - mode d'appel :
iri
 - besoin :
ensemble
 - utilise :
consult_journal, efecran, message, curseur, lirechar, écrit, sc_find_x, sc_find_y, liredate, write, lireentier, erreur, str, insert, lirestring, maj_journal, susp_ao
 - paramètres internes :
 - okset : set of char : ensemble de caractères admissibles
 - car : char : un caractère lu à l'écran
 - reun_inf : boolean : désir de réunion d'information
 - visit_réf : boolean : désir de visite de référence
 - x, y : integer : coordonnées de curseur
 - hh, min : integer : heures et minutes
 - bid : integer : variable bidon
 - lieu_reun : string : lieu de réunion
 - date : string : date de réunion
 - h, m : string : heures et minutes
 - heure : string : horaire
 - fonction :
lecture et stockage des variables de décision, désir de réunion d'information et de visite de référence, si désir de réunion alors lecture et stockage du lieu, de la date et de l'heure de réunion.

67) - nom :
iae
- structure :
procedure
- mode d'appel :
iae
- besoins :
ensemble, cr
- utilise :
consult_journal, efecran, message, curseur, lirechar,
calcster, sc_find_x, sc_find_y, sc_eras_eos, gotoxy,
write, lirestring, escrit, maj_journal, susp_ao
- paramètres internes :
okset, ens_texte : set of char : ensemble de caractères admissibles

car : char : un caractère lu à l'écran

x, y : integer : coordonnées de curseur

z : integer : numéro de ligne
nbre_ligne : integer : nombre de lignes (compteur)

b : integer : variable bidon
ct : boolean : variable pour la BD
etapes : string : étapes
ster : string : = "CR"
- fonction :
lecture et stockage des étapes.

68) - nom :
iqa
- structure :
procedure
- mode d'appel :
iqa
- besoins :
ensemble, cr
- utilise :
consult_journal, liref17, curseur, write, efecran,
message, sc_find_x, sc_find_y, texte_libre, escrit,
maj_journal, calcster
- paramètres internes :
ens_texte : set of char : ensemble de caractères admissibles

x, y : integer : coordonnées de curseur
y2 : integer : numéro de ligne
ci, ci2, l : integer : variables pour la BD
ct, fin : boolean : variables pour la BD
descrip : string : description des questions
ster : string : = "CR"
- fonction :
lecture et stockage des questions relatives aux applications.

69) - nom :
 idac
- structure :
 procedure
- mode d'appel :
 idac
- besoins :
 ensemble, cr
- utilise :
 consult_journal, efecran, message, curseur, ecrittl,
 liref24, sc_find_x, sc_find_y, texte_libre, maj_
 journal, susp_ao
- paramètres internes :
 ens_texte : set of char : ensemble de caractères
 admissibles
 x, y : integer : coordonnées de curseur
 y2 : integer : numéro de ligne
 ct : boolean : variable pour la BD
 descrip : string : description ancienne con-
 figuration
 conf_inf : boolean : existence d'une configura-
 tion informatique
- fonction :
 lecture et stockage de la description de l'ancienne
 configuration si elle existe.

70) - nom :
 verif_j_b
- structure :
 procedure
- mode d'appel :
 verif_j_b
- utilise :
 reset, ioresult, message, lirereturn, close
- paramètres internes :
 fich_journ : f4 : fichier journal de bord
- fonction :
 vérifie l'existence du journal de bord sur la
 disquette de travail, s'il existe un message est
 affiché.

71) - nom :
 susp_ao
- structure :
 procedure
- mode d'appel :
 susp_ao
- utilise :
 message, curseur, lirechar, exit

- besoin :
 routage
- modifie :
 routage
- paramètres internes :
 okset : set of char : ensemble de caractères
 admissibles
- car : char : un caractère lu à l'écran
- fonction :
 lecture du désir ou du refus de suspension.

- 71b) - nom :
 liredate
- structure :
 procédure
 - mode d'appel :
 liredate (x, y, date)
 - entrée :
 x, y : integer : coordonnées de curseur
 - sortie :
 date : string : date lue
 - utilise :
 gotoxy, write, lireentier, erreur, str, insert
 - paramètres internes :
 jj, mm, aa : integer : jour, mois, année
 j, m, a : string : jour, mois, année
 - fonction :
 lecture aux coordonnées "(x,y)" d'une date "date".

- 72) - nom :
 calcster
- structure :
 procédure
 - mode d'appel :
 calcster (s)
 - sortie :
 s : string : = "CR" (carriage return)
 - utilise :
 chr, copy, concat
 - paramètres internes :
 stbid : string : variable intermédiaire
 st : string [1] : variable intermédiaire
 - fonction :
 retourne dans "s" le "CR".

ANNEXE 12.3. : MODULE "INTRODUCTION DES DONNEES L.I."

- 73) - nom :
intro_li
- structure :
procédure
- mode d'appel :
intro_li
- utilise :
welcome_li, idate, i_act, i_applic
- fonction :
gère, au plus haut niveau, le scénario de l'introduction des données de la lettre d'intention.
- 74) - nom :
welc_li
- structure :
procedure
- mode d'appel :
welc_li
- utilise :
efecran, message, time, curseur, lirechar, lirestring, verif_j_b, transf_j_bord, chng_name, lirereturn, verif_disk, exit, init
- besoin :
ndisk
- modifie :
ndisk
- paramètres internes :
okset : set of char : ensemble de caractères admissibles
car : char : un caractère lu à l'écran
nom_cli : string : nom de la disquette de travail
ok : boolean : = true si nom disquette correspond
date : string : date de création L.I.
x, y : integer : coordonnées de curseur
bid : integer : variable bidon
- fonction :
gère les opérations de départ (i.e. si c'est une nouvelle L.I. modifier le nom de la disquette sinon vérifier le nom,...).

- 75) - nom :
 idate
- structure :
 procédure
- mode d'appel :
 idate
- utilise :
 efecran, message, curseur, liredate, escrit,
 sc_find_x, sc_find_y
- paramètres internes :
 x, y : integer : coordonnées de curseur
 bid : integer : variable bidon
 date : string : date de réponse de la L.I.
- fonction :
 lecture et stockage de la lettre de réponse de la
 L.I.

- 76) - nom :
 i_act
- structure :
 procedure
- mode d'appel :
 i_act
- besoins :
 enscar, cr
- utilise :
 efecran, message, curseur, sc_find_x, sc_find_y,
 texte_libre, ecritt1
- paramètres internes :
 ens_texte : set of char : ensemble de caractères
 admissibles

- x, y : integer : coordonnées de curseur
- y2 : integer : numéro de ligne
- bid : integer : variable bidon
- ct : boolean : variable pour la BD
- descrip : string : description de l'activité
- fonction :
 lecture et stockage de la description de l'activité
 de l'entreprise pour la L.I.

ANNEXE 12.4. : MODULE "TOOLKIT"

77) - nom :
menu_toolkit
- structure :
procédure
- mode d'appel :
menu_toolkit
- utilise :
efecran, curseur, lirechar, message, text_data,
annexel, menu_journal, menu_curseur, list_ao,
imp_acc, lec_code, list_li
- paramètres internes :
car : char : un caractère lu à l'écran
okset : set of char : ensemble de caractères
admissibles
- fonction :
propose, d'une manière interactive, de choisir
dans un menu une opération ou un autre menu.
- algorithme
- okset := 'S','M','T','G','J','C','L','I','X',
's','m','t','g','j','c','l','i','x' ;
- répéter
- nettoyer l'écran;
- envoyer un message à l'écran;
- positionner le curseur;
- lire un caractère de okset =: car;
- selon car faire
'T','t' : text_data;
'G','g' : annexel;
'J','j' : menu_journal;
'C','c' : menu_curseur;
'L','l' : list_ao;
'S','s' : list_li;
'I','i' : imp_acc;
'M','m' : lec_code;
fin selon
jusqu'à car = 'X' ou car = 'x';

78) - nom :
trans
- structure :
procédure
- mode d'appel :
trans (fich1, fich2)

- entrée :
 - fich1 : string : nom du fichier texte à transformer
 - fich2 : string : nom du fichier résultat
- utilise :
 - rewrite, reset, close, readln, put, erreur, eof, ioreult
- paramètres internes :
 - s : string : une ligne du fichier texte
 - fich_a_trans : text : fichier texte
 - fich_res : f1 : fichier de string
- fonction :
 - transforme le fichier texte fich1 en un fichier fich2 permettant l'accès direct.

79) - nom :
text_data

- structure :
 - procédure
- mode d'appel :
 - text_data
- besoins :
 - ensemble
- utilise :
 - efecran, gotoxy, write, lirestring, concat, trans, acces_tech
- paramètres internes :
 - okset : set of char : ensemble de caractères admissibles
 - vol1, vol2 : string : numéro de volume
 - nomfich : string : nom du fichier à transformer
 - fichier : string : nom du fichier résultat
- fonction :
 - permet, de manière interactive, de demander les informations nécessaires à l'activation de trans; et l'active.

80) - nom :
crea_an1

- structure :
 - procédure
- mode d'appel :
 - crea_an1 (fich1, fich2)
- entrée :
 - fich1 : string : nom du fichier générateur
 - fich2 : string : nom du fichier annexe
- utilise :
 - reset, ioreult, rewrite, put, eof, get, close, seek; erreur

```
- paramètres internes :
ncase      : integer : numéro de record courant
ntexte    : integer : nombre de messages
switch     : integer : détermine quand on change de
                    message
c          : string  : un élément de fich1
fich_gen  : f1      : fichier de string générateur
fich_anx1 : f2      : fichier d'entier
- fonction :
crée, pour le fichier fich1, un fichier fich2 con-
tenant des "pointeurs" vers chaque message de fich1.
- algorithme
- (x$I-x)
- ouvrir le fichier fich1 pour lecture;
- si le fichier existe
  alors - ouvrir le fichier fich2 pour écriture;
        - ncase := 0; ntexte := 0; switch := 0;
        - fenêtre du fich2 := ntexte;
        - charger un élément de fich2;
        - tantque pas fin fich1 faire
          - charger fenêtre fich 1;
          - c := fenêtre fich 1;
          - ncase := ncase + 1;
          - si c[1] = '$'
            alors switch = 0
            sinon si switch = 0
                  alors - fenêtre fich2 := ncase;
                          - charger un élément de fich2;
                          - ntexte := ntexte + 1;
                          - switch := 1;
                  finsi
            finsi
          finfaire
        - fermer les fichiers fich1 et fich2;
        - ouvrir le fichier fich2 pour lecture;
        - se positionner au 0° record;
        - fenêtre fich2 := ntexte;
        - charger un élément de fich2;
        - fermer le fichier fich2;
  sinon envoyer un message d'erreur;
  finsi
```

```
81) - nom :
      annexel
- structure :
  procedure
- mode d'appel :
  annexel
- besoin :
  ensemble
```


- utilise :
efecran, gotoxy, write, lirestring, concat, crea_an1,
acces_techn
- paramètres internes :
okset : set of char : ensemble de caractères
admissibles
vol1 : string : numéro de volume
nomfich : string : nom du fichier générateur
fichier : string : nom du fichier résultat
- fonction :
permet, de manière interactive, de demander les in-
formations nécessaires à l'activation de crea_an1;
et active crea_an1.

- 82) - nom :
lec_code
- structure :
procedure
 - mode d'appel :
lec_code
 - utilise :
efecran, message, curseur, lirecode, decode,
erreur, exit, encode
 - paramètres internes :
pass : string : valeur encodée du mot de passe
old_pass, new_pass1, new_pass2 : string : volume
du mot de passe
 - fonction :
permet, d'une manière interactive, de modifier le
mot de passe.

- 83) - nom :
acces_techn
- structure :
function : integer
 - mode d'appel :
acces_techn
 - utilise :
efecran, message, curseur, lirecode, decode, erreur
 - paramètres internes :
code : string : mot de passe lu à la keyboard
pass : string : mot de passe encodé
 - fonction :
détermine si l'utilisateur a accès à la fonction
qu'il a choisie, dans le cas favorable acces_techn
= 1 (sinon = 0).

- 84) - nom :
menu_curseur
- structure :
procédure
- mode d'appel :
menu_curseur
- utilise :
acces_techn, efecran, gotoxy, write, lirechar,
ajout_cur, sup_cur, lister_cur, modif_cur
- paramètres internes :
car : char : un caractère lu à l'écran
okset : set of char : ensemble de caractères
admissibles
- fonction :
permet de choisir, d'une façon interactive, une des
4 fonctions ajouter - lister - modifier - supprimer
curseur.
- 85) - nom :
ajout_cur
- structure :
procédure
- mode d'appel :
ajout_cur
- utilise :
init_curseur, reset, efecran, write, gotoxy, bip,
sc_erase_to_eol, lireentier, erreur, close, put,
lire_a_coord, lirechar
- paramètres internes :
num_mat : integer : numéro de matrice
num_mes : integer : numéro de message
car : char : un caractère lu à l'écran
okset : set of char : ensemble de caractères
admissibles
fich_cur : f3 : fichier de matrices
- fonction :
permet, d'une manière interactive, d'ajouter de
nouvelles coordonnées de curseur (par ligne).
- 86) - nom :
lire_a_coord
- structure :
procédure
- mode d'appel :
lire_a_coord (num_mat, num_mes)
- entrée :
num_mat : integer : numéro de matrice concernée
num_mes : integer : numéro de message
- besoin :
moteur
- modifie :
moteur

- utilise :
gotoxy, write, lireentier, bip, erreur, sc_erase_
to_eol, lirechar, enc_int
- paramètres internes :
val : integer : valeur d'un élément de matrice
x, y : integer : coordonnées de curseur
I : integer : indice de matrice (colonne)
car : char : un caractère lu à l'écran
okset : set of char : ensemble de caractères
admissibles
- fonction :
obtient, d'une manière interactive, les coordonnées
de curseur et les insère dans la matrice de numéro
"num_mat" pour le "num_mes"ème message.

- 87) - nom :
modif_cur
- structure :
procedure
 - mode d'appel :
modif_cur
 - utilise :
efecran, gotoxy, write, sc_erase_to_eol, lireentier,
bip, erreur, init_curseur, lire_m_coord, lirechar,
seek, reset, put, close
 - paramètres internes :
num_mat : integer : numéro de matrice
num_mes : integer : numéro de message
car : char : un caractère lu à l'écran
okset : set of char : ensemble de caractères
admissibles

fich_cur : f3 : fichier de matrices
 - fonction :
permet, d'une manière interactive, de modifier des
coordonnées de curseur.

- 88) - nom :
lire_m_coord
- structure :
procedure
 - mode d'appel :
lire_m_coord (num_mat, num_mes)
 - entrée :
num_mat : integer : numéro de la matrice concernée
num_mes : integer : numéro de message
 - besoin :
moteur
 - modifie :
moteur

- utilise :
gotoxy, write, lireentier, bip, erreur, sc_erase_
to_eol, lirechar, enc_int
- paramètres internes :
val : integer : valeur d'un élément de
matrice
x, y : integer : coordonnées de curseur
I : integer : indice de matrice (colonne)
car : char : un caractère lu à l'écran
okset : set of char : ensemble de caractères
admissibles
- fonction :
obtient, d'une manière interactive, les coordonnées
de curseur à modifier et les modifie dans moteur
pour la matrice de numéro "num_mat" et le message
de numéro "num_mes".

- 89) - nom :
- sup_cur
 - structure :
procedure
 - mode d'appel :
sup_cur
 - besoin :
moteur
 - modifie :
moteur
 - utilise :
efecran, gotoxy, write, sc_erase_to_eol, lireentier,
bip, erreur, init curseur, reset, seek, put, close,
lirechar
 - paramètres internes :
I : integer : variable de travail d'une
boucle
num_mat : integer : numéro de matrice
num_mes : integer : numéro de message
car : char : un caractère lu à l'écran
okset : set of char : ensemble de caractères
admissibles
fich_cur : f3 : fichier de matrices
 - fonction :
permet, d'une manière interactive, de supprimer
les coordonnées concernant le message numéro "num_mes"
de la matrice de numéro "num_mat".

90) - nom :
lister_cur
- structure :
procédure
- mode d'appel :
lister_cur
- utilise :
efecran, gotoxy, write, lirechar, openprinter,
prlongueur, ioresult, reset, seek, eof, get,
writeln, dec_int, lirereturn, closeprinter, erreur,
close
- paramètres internes :
x, y : integer : coordonnées de curseur
I, J, K : integer : variables de travail
car : char : un caractère lu à l'écran
okset : set of char : ensemble de caractères
admissibles
fich_cur : f3 : fichier de matrices
- fonction :
permet de lister le contenu des matrices de coordon-
nées à l'écran ou à l'imprimante.

91) - nom :
menu_journal
- structure :
procedure
- mode d'appel :
menu_journal
- utilise :
acces_techn, efecran, message, curseur, lirechar,
ajout_journal, supprimer_journal, lister_journal
- paramètres internes :
car : char : un caractère lu à l'écran
okset : set of char : ensemble de caractères
admissibles
- fonction :
permet de choisir d'une manière interactive, une
des 3 fonctions ajouter - lister - supprimer
journal.

92) - nom :
ajout_journal
- structure :
procedure
- mode d'appel :
ajout_journal
- besoins :
fich_journ, ensemble
- modifie :
fich_journ

```
- utilise :
  reset, close, seek, put, rewrite, efecran, lirechar,
  lirestring, message, curseur, write, reserver,
  ioresult
- paramètres internes :
  k          : integer      : numéro de record libre
  test       : integer      : contient valeur de
                           ioresult
  x, y, x2, y2 : integer    : coordonnées de curseur
  car        : char        : un caractère lu à
                           l'écran
  nomproc    : string      : nom de procédure
  okset      : set of char  : ensemble de caractères
                           admissibles

- fonction :
  permet, d'une manière interactive, d'ajouter une
  nouvelle procédure au journal de bord.
- algorithme
  - okset := 'X','x','C','c' ;
  - (x$I-x)
  - ouvrir le fichier fich_journ;
  - test := ioresult;
  - nettoyer l'écran;
  - envoyer des messages à l'écran;
  - répéter
  - positionner curseur et effacer fin de ligne;
  - lire un string := nomproc;
  - si test = 0
    alors - réserver (k);
            - se positionner au ko record;
            - fich_journ .nom_proc := nomproc;
            - fich_journ .switch_proc := 0;
            - insérer la fenêtre;
            - positionner le curseur;
            - lire un caractère de okset := car;
            - si car = 'X' ou car = 'x' alors fermer
              le fichier
    sinon - ouvrir fichier (création);
            - fich_journ .nom_proc := '2';
            (x = initialisation liste chaînée x)
            - charger un élément du fichier;
            - fenêtre := nomproc et switch à 0;
            - charger un élément du fichier;
            - fenêtre := 'xxxxx', (x = fin du fichier x)
            - charger un élément du fichier;
            - fermer le fichier;
            - test := 0;
            - positionner le curseur;
            - lire un caractère de okset := car;
            - fermer le fichier;
            - si car = 'C' ou car = 'c'
              alors ouvrir le fichier fich_journ
    jusqu'à car = 'X' ou car = 'x';
  (x$I+x)
```


- 93) - nom :
supprimer_journal
- structure :
procédure
- mode d'appel :
supprimer_journal
- besoin :
fich_journal
- modifiée :
fich_journal
- utilise :
reset, close, seek, put, efecran, lireentier, message,
curseur, write, liberer, lirechar, erreur, ioresult
- paramètres internes :
ln : integer : numéro de record à libérer
num_proc : integer : numéro de procédure
car : char : un caractère lu à l'écran
okset : set of char : ensemble de caractères
admissibles
- fonction :
permet, d'une manière interactive, de supprimer
une procédure du journal de bord.
- 94) - nom :
lister_journal
- structure :
procédure
- mode d'appel :
lister_journal
- besoin :
fich_journal
- utilise :
reset, close, get, write, curseur, lirereturn,
message, lirechar, writeln, seek, efecran, openprinter,
prlongueur, ioresult, closeprinter
- paramètres internes :
nbre_proc : integer : nombre de procédures
car : char : un caractère lu à l'écran
okset : set of char : ensemble de caractères
admissibles
- fonction :
permet de lister le contenu du journal de bord
à l'écran ou à l'imprimante.
- 95) - nom :
réservé
- structure :
procédure
- mode d'appel :
réservé (h)

XK

```
- entrée :
  k : integer : numéro du 1° record libre
- besoin :
  fich_journ
- modifie :
  fich_journ
- utilise :
  seek, get, put, strtoint, str
- paramètres internes :
  k2      : integer : variable de travail
  st1, st2 : string  : valeur d'un champ du record
- fonction :
  indique par k la première place libre dans le fichier
  journal de bord.
- algorithme
  - se positionner au record 0;
  - charger la fenêtre;
  - fenêtre =: st1; (x contient 1° place de la liste
  chaînée x)
  - transformer st1 en un entier =: k;
  - se positionner au record k;
  - charger la fenêtre;
  - si fenêtre 'xxxxx'
    alors - se positionner au record k;
          - charger la fenêtre;
          - fenêtre =: st2;
          - transformer st2 en un entier =: k2;
          - si k2 0
            alors - se positionner au record 0;
                  - st2 =: fenêtre;
                  - charger un élément du fichier;
          (x cas où on n'est pas en fin de fichier x)
    sinon - 'xxxxx' =: fenêtre;
          - charger un élément du fichier;
          - k2 := k + 1;
          - se positionner au record 0;
          - transformer k2 en st2;
          - fenêtre := st2;
          - charger un élément du fichier.
```

```
96) - nom :
      liberer
- structure :
  procedure
- mode d'appel :
  liberer (k)
- entrée :
  k : integer : numéro de record à libérer
- besoin :
  fich_journ
- modifie :
  fich_journ
```


- utilise :
seek, get, put, str
- paramètres internes :
st, st2 : string : valeur d'un champ du record
- fonction :
libère le k^o record dans le fichier journal de bord.
- algorithme
 - se positionner au record 0;
 - charger la fenêtre;
 - fenêtre =: st;
 - transformer k en un string =: st2;
 - se positionner au record 0;
 - fenêtre := st2;
 - charger un élément du fichier;
 - se positionner au record k;
 - fenêtre := st;
 - charger un élément du fichier;
 - = mise à jour de la liste chaînée.

96b) - nom :
list_ao

- structure :
procedure
- mode d'appel :
list_ao
- utilise :
reset, ioresult, close, impacc, erreur
- paramètres internes :
fichtxt : text : fichier texte
- fonction :
imprime le texte de l'appel d'offres.

96c) - nom :
list_li

- structure :
procedure
- mode d'appel :
list_li
- utilise :
reset, ioresult, close, impacc, erreur
- paramètres internes :
fichtxt : text : fichier texte
- fonction :
imprime la lettre d'intention.

- 96d) - nom :
 - imp_ac
- structure :
 - procedure
- mode d'appel :
 - imp_ac
- utilise :
 - extérieur
- fonction :
 - permet d'imprimer un fichier suite à des questions en interaction avec l'utilisateur.

ANNEXE 12.5. : MODULE "HELP"

- 97) - nom :
help_system
- structure :
procedure
- mode d'appel :
help_system
- utilise :
efecran, message, lirereturn
- fonction :
affiche à l'écran le mode d'emploi du logiciel.
- 98) - nom :
help_i_texte
- structure :
procedure
- mode d'appel :
help_i_texte
- utilise :
efecran, message, lirereturn
- fonction :
affiche à l'écran le mode d'emploi d'introduction
des textes.
- 99) - nom :
help_editeur
- structure :
procedure
- mode d'appel :
help_editeur
- utilise :
efecran, message, lirereturn
- fonction :
affiche à l'écran le mode d'emploi de l'éditeur
de textes.
- 100) - nom :
help_cod
- structure :
procedure
- mode d'appel :
help_cod

- utilise :
efecran, message, curseur, lirereturn
- fonction :
affiche à l'écran une aide pour le codage.

- 101) - nom :
help_ev
- structure :
procedure
 - mode d'appel :
help_ev
 - utilise :
efecran, message, curseur, lirereturn
 - fonction :
affiche à l'écran une aide pour le menu principal.

ANNEXE 12.6. : MODULE "PRESENTATION DE TABLEAUX
COMPARATIFS"

- 102) - nom :
tab_impr
- structure :
procedure
- mode d'appel :
tab_impr (fichier, choix)
- entrée :
fichier : string : nom du fichier à imprimer
choix : char : détermine où il faut l'imprimer
- utilise :
message, lirereturn, openprinter, efecran,
concat, copy, length, reset, eof, readln, writeln,
gotoxy, write, close, closeprinter
- paramètres internes :
fich : text : fichier texte à imprimer
s : string : variable de travail
compt : integer : compteur de ligne à l'écran
i : integer : variable de boucle
- fonction :
imprime le fichier texte de nom "fichier" à l'écran
ou à l'imprimante selon la valeur de "choix"
(E, e, I, i).
- 103) - nom :
tab_sup
- structure :
procedure
- mode d'appel :
tab_sup
- utilise :
d_rem_files, d_krunch
- paramètres internes :
result : d_result : résultat des appels systèmes
- fonction :
supprimer les fichiers des tableaux et réorganiser
la disquette de travail.

104) - nom :
tab_num
- structure :
procédure
- mode d'appel :
tab_num
- besoin :
rac2
- utilise :
efecran, message, curseur, lirechar, reset,
tab_reorg, ioresult, rewrite, t_e_quest, t_e_crit,
t_e_syn, writeln, tab_fo, close, tab_impr
- paramètres internes :
okset : set of char : ensemble de caractères
admissibles
car : char : un caractère lu à l'écran
choix : char : détermine le choix de fi-
chier à lire
fichier : string : nom de fichier
i, j : integer : variables de boucle
ci : integer : détermine l'ouverture ou
la fermeture de fichiers
boucl : integer : compteur de questions (ou
de critères)
num_rec : integer : numéro d'enregistrement
num : integer : numéro de questions (ou de
critères)
long : integer : nombre de lignes des ques-
tions (ou des critères)
lg : integer : nombre de questions (ou
de critères)
tableaulu : tab20 : texte d'une question (ou
d'un critère)
fin : boolean : fin du fichier
fich : text : fichier texte
- fonction :
crée le tableau comprenant le numéro des questions
et leur texte (idem pour les critères), le numéro
et le nom des fournisseurs; ensuite, l'imprime
ou l'affiche à l'écran.

105) - nom :
t_e_quest
- structure :
procédure
- mode d'appel :
t_e_quest
- besoins :
fich, boucl, j, choix, ci, i, num_rec, num, long,
tableaulu, fin, lg

- modifie :
fich, boucl, i, j, ci, choix, num_rec, num, long,
tableaulu, fin, lg
- utilise :
writeln, liregen
- fonction :
écrit dans le fichier "fich" les informations con-
cernant les questions (numéro dans le tableau,
numéro et texte).
- commentaire :
fait partie de "tab_num".

106) - nom :
t_e_crit

- structure :
procedure
- mode d'appel :
t_e_crit
- besoins :
fich, boucl, j, choix, ci, i, num_rec, num, long,
tableaulu, fin, lg
- modifie :
fich, boucl, j, choix, ci, i, num_rec, num, long,
tableaulu, fin, lg
- utilise :
writeln, liregen
- fonction :
écrit dans le fichier "fich" les informations con-
cernant les critères (numéro dans le tableau,
numéro et texte).
- commentaire :
fait partie de "tab_num".

107) - nom :
t_e_syn

- structure :
procedure
- mode d'appel :
t_e_syn
- besoins :
fich, boucl, choix, ci, i, num_rec, num, long,
tableaulu, fin, lg
- modifie :
fich, boucl, choix, ci, i, num_rec, num, long,
tableaulu, fin, lg
- utilise :
writeln, liregen

- fonction :
écrit dans le fichier "fich" les informations concernant les questions synoptiques (numéro dans le tableau, numéro et texte).
- commentaire :
fait partie de "tab_num".

- 108)
- nom :
tab_fo
 - structure :
procédure
 - mode d'appel :
tab_fo (arbre)
 - entrée :
arb : pointalp : arbre des fournisseurs
 - besoins :
fich, sw_arb_f_ex
 - modifie :
fich, sw_arb_f_ex
 - utilise :
tab_fo, writeln, crea_arb_f
 - fonction :
écrit dans le fichier "fich" les informations concernant les fournisseurs (numéro et nom).
 - commentaire :
fait partie de "tab_num".

- 109)
- nom :
tab_mat
 - structure :
procédure
 - mode d'appel :
tab_mat
 - utilise :
efecran, message, curseur, lirechar, reset, ioresult, tab, crea, close, tab_impr, tab_reorg
 - paramètres internes :
okset : set of char : ensemble de caractères admissibles
 - car : char : un caractère lu à l'écran
 - fichi : string : nom de fichier
 - fich : text : fichier texte
 - fonction :
gère la création et l'impression du tableau concernant les questions synoptiques.

- 110) - nom :
tab_crit
- structure :
procédure
- mode d'appel :
tab_crit
- utilise :
efecran, message, curseur, lirechar, reset, close,
ioresult, tab_crea, tab_impr, tab_reorg
- paramètres internes :
okset : set of char : ensemble de caractères
admissibles
car : char : un caractère lu à l'écran
fichi : string : nom de fichier
fich : text : fichier texte
- fonction :
gère la création et l'impression du tableau con-
cernant les critères.
- 111) - nom :
tab_quest
- structure :
procédure
- mode d'appel :
tab_quest
- utilise :
efecran, message, curseur, lirechar, reset, close,
ioresult, tab_crea, tab_impr, tab_reorg
- paramètres internes :
okset : set of char : ensemble de caractères
admissibles
car : char : un caractère lu à l'écran
fichi : string : nom de fichier
fich : text : fichier texte
- fonction :
gère la création et l'impression du tableau con-
cernant les questions.
- 112) - nom :
tab_fin
- structure :
procédure
- mode d'appel :
tab_fin
- utilise :
efecran, message, curseur, lirechar, reset, close
ioresult, tab_crea, tab_impr, tab_reorg

- paramètres internes :
 - okset : set of char : ensemble de caractères admissibles
 - car : char : un caractère lu à l'écran
 - fichi, fichi2, fichi3 : string : noms de fichiers
 - fich, fich2, fich3 : text : fichiers texte
- fonction :
 - gère la création et l'impression des tableaux concernant les questions, les critères et les questions synoptiques.

- 113) - nom :
tableaux
- structure :
procédure
 - mode d'appel :
tableaux
 - utilise :
efecran, message, curseur, lirechar, tab_num, tab_sup, tab_quest, tab_crit, tab_mat, tab_fin
 - paramètres internes :
 - okset : set of char : ensemble de caractères admissibles
 - car : char : un caractère lu à l'écran
 - fonction :
 - permet, de manière interactive, de choisir une fonction relative à l'affichage (ou impression) de tableaux.

- 113b) - nom :
tab_reorg
- structure :
procédure
 - mode d'appel :
tab_reorg
 - besoins :
fich_coms
 - utilise :
reset, f_start, f_lengtt, d_krunch, close
 - paramètres internes :
 - debut : integer : numéro de bloc de départ du fichier des commentaires
 - long : integer : longueur (en bloc) du fichier des commentaires
 - total : integer : début du bloc autour duquel on réorganise la disquette
 - result : d_result : résultat de d_krunch
 - fonction :
 - réorganise la disquette de travail.

ANNEXE 12.7. : MODULE "CODAGE DES REPONSES"

- 114) - nom :
susp_ev
- structure
fonction (boolean)
- mode d'appel :
susp_ev
- utilise :
message, curseur, lirechar
- paramètres internes :
okset : set of char : ensemble de caractères
admissibles
car : char : un caractère lu à l'écran
- fonction :
permet de suspendre ou de continuer le codage
des réponses aux questions.
- 115) - nom :
augm_compt
- structure :
procédure
- mode d'appel :
augm_compt (choice, rep)
- entrée :
choice : char : détermine le type des réponses
rep : string : codage d'une réponse
- besoin :
com_ev
- modifie :
com_ev
- fonction :
augmente les compteurs des réponses suivant d'une
part les réponses "rep" et d'autre part le type
des réponses "choic"
où choice = - Q : relatif aux questions
- C : relatif aux critères
- Y : relatif aux questions synoptiques
rep peut avoir comme valeur :
0, o, Ox, ox, N, n, Nx, nx, x, -.

116) - nom :
maj_com
- structure :
procédure
- mode d'appel :
maj_com (i)
- entrée :
i : integer : numéro du vecteur à mettre à 1
- besoin :
fich_com
- modifiée :
fich_com
- utilise :
reset, seek, put, close
- fonction :
met à 1 l'élément "i" du vecteur des commandes.

117) - nom :
dep_quest
- structure :
procédure
- mode d'appel :
dep_quest
- besoin :
sel_enreg
- utilise :
efecran, curseur, sc_find_x, sc_find_y, cut_dec_fx,
erreur, dern_q, liregen, message, gotoxy, write,
ecriecran, lirecodage, augm_compt, lirecom, sauv_
rep, sc_erase_to_eol, susp_ev, maj_com
- paramètres internes :
x1, y1 : integer : coordonnées de curseur
x2, y2 : integer : coordonnées de curseur
x3, y3 : integer : coordonnées de curseur
x4, y4 : integer : coordonnées de curseur
ci, ci2 : integer : détermine ouverture ou ferme-
ture de fichier
i : integer : numéro d'enregistrement
j : integer : variable de boucle
num_rec : integer : numéro d'enregistrement
num : integer : numéro de question
n : integer : numéro d'enregistrement
long : integer : nombre de lignes d'une question
boucl : integer : compteurs de questions
l : integer : longueur des commentaires
lg : integer : nombre de questions
repo : string : codage d'une réponse
tableaulu : tab20 : texte d'une question
coms : tab3 : texte d'un commentaire
fin : boolean : fin du fichier i consulter
stop : boolean : est à vrai lorsque l'offre
est rejetée

susp : boolean : est à vrai lorsqu'en suspend
patience : boolean : est à vrai lorsqu'il faut
se positionner dans le fichier
des questions
choix : char : caractère permettant de dé-
terminer s'il s'agit des
questions informationnelles
ou des questions personali-
sées

- fonction :
permet, de manière interactive, de coder les répon-
ses aux questions personnalisées et aux questions
informationnelles; et, ce avec possibilité de sus-
pendre et de reprendre.

118) - nom :
an_crit
- structure :
procedure
- mode d'appel :
an_crit
- besoin :
sel_enreg
- utilise :
efecran, curseur, sc_find_x, sc_find_y, liregen,
message, write, gotoxy, ecriecran, lirecodage,
augm_compt, verif_crit, maj_crit_dec, lirecom,
sauv_rep, maj_com, maj_crit_f
- paramètres internes :
x1, y1 : integer : coordonnées de curseur
x2, y2 : integer : coordonnées de curseur
x3, y3 : integer : coordonnées de curseur
x4, y4 : integer : coordonnées de curseur
ci, ci2 : integer : détermine l'ouverture ou la
fermeture de fichier

i : integer : numéro d'enregistrement
j : integer : variable de boucle
num_rec : integer : numéro d'enregistrement
n : integer : numéro d'enregistrement
num : integer : numéro de critère
long : integer : nombre de lignes d'un critère
boucl : integer : compteur de critères
l : integer : nombre de lignes d'un commen-
taire

lg : integer : nombre de critères
repo : string : codage de satisfaction d'un
critère

tableaulu : tab20 : texte d'un critère
coms : tab3 : texte d'un commentaire
fin : boolean : fin du fichier à consulter

stat : boolean : statut d'un critère
stop : boolean : est à vrai lorsque l'offre
est rejetée
choix : char : caractère permettant de déter-
miner s'il s'agit des critères
supplémentaires ou des critè-
res personnalisés

- fonction :
permet, de manière interactive, de coder la satis-
faction des critères personnalisés et des critères
supplémentaires.

119) - nom :
stat_crit
- structure :
procédure
- mode d'appel :
stat_crit
- utilise :
efecran, message, curseur, sc_find_x, sc_find_y,
liregen, sc_erase_to_eol, gotoxy, ecriecran,
lirechar, sauv_stat, maj_com
- paramètres internes :
x1, y1 : integer : coordonnées de curseur
x2, y2 : integer : coordonnées de curseur
x3, y3 : integer : coordonnées de curseur
i : integer : variable de boucle
ci, ci2 : integer : détermine l'ouverture ou la
fermeture de fichier
num : integer : numéro de critère
num_rec : integer : numéro d'enregistrement
long : integer : nombre de lignes d'un critère
boucl : integer : compteur de critères
lg : integer : nombre de critères
okset : set of char : ensemble de critères
admissibles
car : char : un caractère lu à l'écran
choix : char : détermine le type des cri-
tères à lire
tableaulu : tab20 : texte d'un critère
fin : boolean : fin du fichier à consulter
stat : boolean : statut d'un critère
- fonction :
permet, de manière interactive, de donner le statut
de critères décisifs à certains critères.

120) - nom :
pres_mat
- structure :
procedure
- mode d'appel :
pres_mat
- besoin :
sel_enreg
- utilise :
efecran, curseur, message, sc_find_x, sc_find_y,
crit_dec_fx, erreur, lire_conf_sup, bip, liregen,
lirechar, gotoxy, write, ecriecran, lirecodage,
augm_compt, lirecom, sauv_rep, maj_conf_sup,
maj_com
- paramètres internes :
x1, y1 : integer : coordonnées de curseur
x2, y2 : integer : coordonnées de curseur
x3, y3 : integer : coordonnées de curseur
x4, y4 : integer : coordonnées de curseur
ci, ci2 : integer : détermine l'ouverture ou la
fermeture de fichier
i : integer : numéro d'enregistrement
n : integer : numéro d'enregistrement
num_rec : integer : numéro d'enregistrement
num : integer : numéro de question
long : integer : nombre de lignes d'une question
boucl : integer : nombre de questions
l : integer : longueur des commentaires
lg : integer : nombre de questions
repo : string : codage d'une réponse
tableaulu : tab20 : texte d'une question
coms : tab3 : texte d'un commentaire
fin : boolean : fin du fichier à consulter
choix : char : un caractère lu à l'écran
okset : set of char : ensemble de caractères
admissibles
- fonction :
permet, de manière interactive, de coder les ré-
ponses aux questions concernant la configuration
informatique.

121) - nom :
dtm_ens_cod
- structure :
procedure
- mode d'appel :
dtm_ens_cod (okset)
- sortie :
okset : set of char : ensemble de caractères
admissibles

- besoins :
fich_com, sel_enreg
 - utilise :
reset, lire_crit_f
 - fonction :
détermine l'ensemble des caractères admissibles
pour le menu codage.
- 122) - nom :
dtm_ens_menu
- structure :
procedure
 - mode d'appel :
dtm_ens_menu (okset)
 - sortie :
okset : set of char : ensemble de caractères
admissibles
 - besoins :
fich_com, sw_com_s
 - modifie :
fich_com
 - utilise :
reset, ioreult, close, rewrite, put
 - paramètres internes :
i : integer : variable de boucle
 - fonction :
détermine l'ensemble des caractères admissibles
pour le menu principal.
- 123) - nom :
codage
- structure :
procedure
 - mode d'appel :
codage
 - besoin :
sel_enreg
 - utilise :
lire_compt, efecran, message, curseur, dtm_ens_cod,
lirechar, an_crit, dep_quest, stat_crit, pres_mat,
help_cod, sauver_compt
 - paramètres internes :
okset : set of char : ensemble de caractères
admissibles
car : char : un caractère lu à l'écran
 - fonction :
propose, de manière interactive, de choisir dans
un menu une opération relative au codage.

124) - nom :
select
- structure :
procedure
- mode d'appel :
select
- besoins :
sw_arb_f_ex, rac2, sel_enreg
- modifie :
sw_arb_f_ex, sel_enreg
- utilise :
efecran, message, crea_arb_f, curseur, sc_find_x,
sc_find_y, list_sel_f, sc_erase_to_eol, gotoxy,
lirestring, f_deja_intro, bip, erreur
- paramètres internes :
x, y : integer : coordonnées de curseur
comp : integer : compteur (3 essais maximum)
nom : string : nom d'un fournisseur
selection : boolean : est à vrai lorsqu'un four-
nisseur a été sélectionné
- fonction :
permet, de manière interactive, de sélectionner
le fournisseur pour lequel on va coder les ré-
ponses ou la satisfaction des critères.

125) - nom :
tr_fourn
- structure :
procedure
- mode d'appel :
tr_fourn
- besoin :
fich_com
- modifie :
fich_com
- utilise :
verif_fourn, efecran, message, curseur, erreur,
lirereturn, gotoxy, write, lirefourn, ecrfourn,
reset, seek, put, close, initcomp, initerit,
init_confs, d_krunch, lirechar
- paramètres internes :
ci, ci2 : integer : détermine l'ouverture ou
la fermeture de fichier
num_rec_f : integer : numéro d'enregistrement de
fournisseur
compt : integer : compteurs de fournisseurs
boucl : integer : compteur
okset : set of char : ensemble de caractères
admissibles
d_fourn : r8 : enregistrement fournisseur

switch_bd : boolean : détermine la fin du fichier
trans : boolean : détermine si un transfert
 a été effectué
car : char : un caractère lu à l'écran
result : d_result: résultat donné par d_krunch
- fonction :
permet, de manière interactive, de transférer
les fournisseurs auxquels on a envoyé un appel
d'offres sur la disquette de travail de
l'évaluation.

126) - nom :
tr_quest
- structure :
procédure
- mode d'appel :
tr_quest
- besoins :
ndisk, fich_com
- modifie :
ndisk, fich_com
- utilise :
verif, efecran, message, curseur, lirereturn,
gotoxy, write, trfs, transf, reset, seek, put,
close, liregen, init_rep, d_krunch, erreur
- paramètres internes :
num_rec : integer : numéro d'enregistrement
num : integer : numéro de question
long : integer : nombre de questions
ci : integer : détermine l'ouverture ou la
 fermeture de fichiers
boucl : integer : compteur
c : integer : compteurs (3 essais maximum)
tableaulu : tab20 : texte de question
fin : boolean : fin du fichier à lire
s : string : chaîne de caractères de travail
result : d_result: résultat donné par d_krunch
- fonction :
gère, de manière interactive, le transfert du
fichier des questions relatives à une affaire
et des questions informationnelles vers la
disquette de travail pour l'évaluation.

127) - nom :
tr_crit
- structure :
procédure
- mode d'appel :
tr_crit

- besoins :
ndisk, fich_com
- modifiée :
ndisk, fich_com
- utilise :
verif, efecran, message, curseur, lirereturn,
gotoxy, write, trfs, transf, reset, seek, put,
close, d_krunch, erreur
- paramètres internes :
c : integer : compteurs d'essais (3 au maximum)
result : d_result : résultat donné par la fonction
d_result
- fonction :
gère, de manière interactive, le transfert du fi-
chier des critères relatifs à une affaire et des
critères supplémentaires vers la disquette de
travail pour l'évaluation.

128) - nom :
transfert

- structure :
procedure
- mode d'appel :
transfert
- utilise :
efecran, message, curseur, lirechar, gotoxy, write,
tr_four, tr_quest, tr_crit, lirereturn
- paramètres internes :
okset : set of char : ensemble de caractères
admissibles
car : char : un caractère lu à l'écran
- fonction :
propose, de manière interactive, de choisir dans
un menu une opération relative au transfert des
fichiers nécessaires.

129) - nom :
menu_eval

- structure :
procedure
- mode d'appel :
menu_eval
- besoin :
sw_com_s
- modifiée :
sw_com_s
- utilise :
efecran, message, curseur, lirechar, dtm_ens_menu,
transfert, select, tableaux, codage, det_arb_f,
help_ev

- paramètres internes :
 - okset : set of char : ensemble de caractères
admissibles
 - car : char : un caractère lu à l'écran
- fonction :
propose, de manière interactive, de choisir une
opération dans le menu principal.

ANNEXE 12.8. : MODULE "RECOUVREMENT ERREUR"

- 130) - nom :
f_deja_intro
- structure :
fonction : boolean
- mode d'appel :
f_deja_intro (nom_f)
- entrée :
nom_f : string : nom d'un fournisseur
- besoin :
rac
- paramètre interne :
existe : boolean : = vrai si le fournisseur existe
- fonction :
vérifie l'existence, dans l'arbre des fournisseurs,
du fournisseur de nom "nom_f", dans ce cas
f_deja_intro = 0 (sinon = 1).
- 131) - nom :
f_existe
- structure :
procedure
- mode d'appel :
f_existe (nom_f, arb, existe)
- entrée :
nom_f : string : nom d'un fournisseur
arb : pointalp : arbre des fournisseurs
- sortie :
existe : boolean : = vrai si le fournisseur existe
- fonction :
parcourt l'arbre "arb" à la recherche du four-
nisseur de nom "nom_f", s'il existe alors
existe = true (sinon = false).

ANNEXE 12.9. : MODULE "CREATION DE TABLEAUX"

132) - nom :
tab_crea

- structure :
procEDURE

- mode d'appel :
tab_crea (choix)

- entrée :
choix : char : détermine le type de tableaux à créer

- besoin :
fich_coms

- utilise :
message, tab, choix, reset, rewrite, writeln,
tabf, tab_f_lig, tab_q_lig, close, tabcomp,
transcom

- paramètres internes :
fichier1, fichier2, fichier3, fichier4, fichier5
: string : noms de fichiers
titre, titre2 : string : titre en entête des fichiers
fich1, fich2, fich3, fich4
: text : fichiers texte
i, j, k, m, n, integer : variables de boucle
l : integer : longueur d'un commentaire
nbreq : integer : nombre de questions
numq : integer : numéro de question
num : integer : numéro d'enregistrement
S : packed array 1..60 of char : une
ligne d'un fichier texte
code : char : code d'une réponse
tab4 : car4 : tableau de travail
tab5 : car5 : tableau de travail
tab6 : car6 : tableau de travail
tab10, tab10b, tab10c : packed array 1..10 of
integer : le "iè" entier du tableau
signifie s'il est à 1 que
le iè fournisseur est à
prendre en compte

- fonction :
crée les tableaux résultats suivant la valeur
de "choix" :
- Q : tableaux relatifs aux questions
- C : tableaux relatifs aux critères
- Y : tableaux relatifs aux questions synoptiques

133) - nom :
tab_choix
- structure :
procédure
- mode d'appel :
tab_choix
- besoins :
choix, titre, fichier1, fichier2, fichier3,
fichier4, fichier5
- modifie :
titre, fichier1, fichier2, fichier3, fichier4,
fichier5
- fonction :
permet de déterminer, selon la valeur de "choix",
le nom des fichiers nécessaires à l'élaboration
de tableaux;
où "choix" :
- Q : tableaux relatifs aux questions
- C : tableaux relatifs aux critères
- Y : tableaux relatifs aux questions synoptiques
- commentaire :
fait partie de "tab_crea".

134) - nom :
tabf
- structure :
procédure
- mode d'appel :
tabf
- besoins :
i, tab10, car, s, nbreq, numq, k, j, fich1, tab10b
- modifie :
i, j, k, car, s, tab10, nbreq, numq, tab10b
- utilise :
read, readln, tab4_dec
- fonction :
permet de remplir le tableau "tab10" où, si le ième
élément vaut 1, cela signifie que le ième fournis-
seur est à prendre en compte.
- commentaire :
fait partie de "tab_crea".

135) - nom :
tab_f_lig
- structure :
procédure
- mode d'appel :
tab_f_lig

- besoins :
i, tab5, tab10, fich2
- modifie :
i, tab5, tab10, fich2
- utilise :
write, tabf_enc
- fonction :
écrit, dans le fichier contenant le tableau,
la ligne comportant le numéro des fournisseurs.
- commentaire :
fait partie de "tab_crea".

136) - nom :
tab_q_lig

- structure :
procedure
- mode d'appel :
tab_q_lig
- besoins :
i, j, k, m, n, nbreq, car, s, tab5, tab6, tab10,
code, l, num, tab4, fich2, fich_coms, fich3
- modifie :
i, j, k, m, n, car, s, tab5, tab6, tab10, tab4,
fich2, fich3
- utilise :
read, readln, tabq_enc, write, tab_dec, tabc_enc,
seek, get, writeln
- fonction :
écrit, dans le fichier contenant le tableau, la
ligne comportant les codes pour toutes les ques-
tions; et, dans le fichier contenant les commen-
taires, le numéro de question, le numéro de four-
nisseur et le commentaire (s'il existe).
- commentaire :
fait partie de "tab_crea".

137) - nom :
tabt_enc

- structure :
procedure
- mode d'appel :
tabt_enc (choice, com, t4)
- entrée :
choice : char : détermine le compteur
com : enr : enregistrement des compteurs
- sortie :
t4 : car4 : tableau contenant les valeurs du
compteur

- utilise :
tabf_enc
- paramètres internes :
bid : integer : contient la valeur du compteur
sbid : string : variable de travail
- fonction :
d'après "choice", détermine le compteur appartenant à "com" et à insérer dans le tableau "t4"
où choice = :
 - O : compteur de "oui"
 - N : compteur de "non"
 - R : compteur de "pas de réponse"
- commentaire :
fait partie de "tab_crea".

138) - nom :
tab_e_c

- structure :
procédure
- mode d'appel :
tab_e_c (choice)
- entrée :
choice : char : détermine le choix des compteurs
- besoins :
tab10b, com_ev, tab4, fich2
- modifie :
tab10b, tab4, fich2
- utilise :
lire_compt, tabt_enc, write, writeln
- paramètres internes :
j : integer : variable de boucle
- fonction :
écrit dans le fichier "fich2" la ligne comportant les valeurs des compteurs déterminés par "choice",
où choice = :
 - Q : compteurs relatifs aux questions
 - C : compteurs relatifs aux critères
 - Y : compteurs relatifs aux questions synoptiques.
- commentaire :
fait partie de "tab_crea".

139) - nom :
tabcomp

- structure :
procédure
- mode d'appel :
tabcomp
- besoins :
fich2, tab10b

- modifie :
fich2
- utilise :
write, writeln, tab_fc_lig, tab_e_c
- fonction :
écrit, dans le fichier contenant le tableau, le tableau relatif aux compteurs.
- commentaire :
fait partie de "tab_crea".

140) - nom :
transcom

- structure :
procedure
- mode d'appel :
transcom
- utilise :
reset, rewrite, eof, readln, writeln, close
- besoins :
fich3, fich4, fichier4, fichier5
- modifie :
fich4
- fonction :
transfère le fichier de nom "fichier 5" du disque virtuel vers la disquette de travail.
- commentaire :
fait partie de "tab_crea".

140b) - nom :
tab_of_rej

- structure :
procedure
- mode d'appel :
tab_of_rej
- besoins :
i, fich 2
- modifie :
i, fich 2
- utilise :
writeln, crit_dec_fx
- fonction :
écrit dans le fichier le numéro des fournisseurs dont l'offre est rejetée.
- commentaire :
fait partie de "tab_crea".

140c) - nom :
tab_fc_lig
- structure :
procedure
- mode d'appel :
tab_fc_lig
- besoins :
i, tab5, tab10b, fich2
- modifie :
i, tab5, tab10b, fich2
- utilise :
write, tabf_enc
- fonction :
écrit, dans le fichier contenant le tableau, la
ligne comportant le numéro des fournisseurs pour
lesquels une colonne au moins du tableau existe.
- commentaire :
fait partie de "tab_crea".

ANNEXE 12.10. : MODULE "CREATION DU FICHIER RESULTAT"

- 141) - nom :
fichresulao
- structure :
procedure
- mode d'appel :
fichresulao
- besoins :
sour : text : fenêtre du fichier source
resul : text : fenêtre du fichier résultat
lig_lu : string : ligne lue
questmod : text : fichier source
textes : text : fichier source
- utilise :
lire
- paramètres internes :
nom, nom1 : string : nom des fichiers
- fonction :
ouvrir les fichiers sources et faire la première lecture.
- 142) - nom :
info
- structure :
procedure
- mode d'appel :
info
- utilise :
txlb, tabl, quest, insf1, insf2, insf6, insf7,
insf11, insf12, insf13, insf16, insf17, insf21,
insf22, insf23, insf25
- paramètres internes :
lo : integer : longueur d'un tableau de string
i : integer : entier à lire
nu : integer : le numéro de la variable
nudiv : integer : partie entière de nu divisé par 100
direct : integer : guide
trav : string : information lue
tabla : tabla : description lue
- fonction :
diriger vers les différentes procédures d'insertion de données dans le texte.

143) - nom :
fin_lig
- structure :
procEDURE
- mode d'appel :
fin_lig (r, no)
- sortie :
r : boolean : indique si la ligne est > 0
no : integer : indique la longueur de la ligne
- besoin :
lig_res : string : la ligne à écrire
- fonction :
indiquer la longueur d'une ligne et si elle est
supérieur à 0.

144) - nom :
ecrire
- structure :
procEDURE
- mode d'appel :
ecrire (ecri)
- entrée/sortie :
ecri : string : la ligne à écrire dans 'resul'
- besoin :
resul : text : idem
- fonction :
écrire une ligne et la vider.

145) - nom :
lire
- structure :
procEDURE
- mode d'appel :
lire (lir)
- sortie :
lir : string : la valeur lue
- besoin :
sour : text : idem
- fonction :
lire une ligne dans le fichier source

146) - nom :
tabl
- structure :
procEDURE
- mode d'appel :
tabl (tableaulu, long)

- entrée :
 - tableau : tab : le tableau à écrire
 - long : integer : la longueur du tableau
- besoin :
 - lig_res : string : idem
- utilise :
 - ecrire
- paramètre interne :
 - i : integer : indice de boucle
- fonction :
 - écrire un tableau.

148) - nom :
txlb

- structure :
 - procedure
- mode d'appel :
 - txlb (numero)
- entrée :
 - numero : integer : numéro du texte à insérer dans le fichier résultat
- besoin :
 - lig_res : string : idem
- utilise :
 - ecrire, liretl
- paramètres internes :
 - ligne : string : la ligne à insérer
 - fintext : boolean : indicateur de fin de texte
- fonction :
 - insérer un texte libre dans un texte final.

149) - nom :
quest

- structure :
 - procedure
- mode d'appel :
 - quest
- besoin :
 - lig_res : string : idem
- utilise :
 - ecrire, tabl, liref4
- paramètres internes :
 - qu : integer : numéro de question
 - i : integer : commande l'ouverture du fichier
 - lo : integer : longueur d'une question
 - j : integer : indice de boucle
 - r : integer : nombre maximum de questions
 - question : tab : texte de la question
 - finquest : boolean : indicateur de fin de fichier
 - trav : string : idem
- fonction : lire et écrire les questions.

- 150) - nom :
soulig
- structure :
procEDURE
- mode d'appel :
soulig (sou1, durée)
- entrée :
sou1 : char : caractère avec lequel il faut
souligner
duree : integer : le nombre de caractères à
souligner
- besoin :
lig_res
- utilise :
ecrire
- paramètres internes :
i : integer : indice de boucle
sou : string : construire le string de souligne-
ment
- fonction :
souligner.
- 151) - nom :
decalage
- structure :
procEDURE
- mode d'appel :
decalage (deca, place)
- entrée :
deca : string : le string à décaler et à écrire
place : integer : indique jusqu'où il faut décaler
- besoin :
lig_res : string : idem
- utilise :
ecrire
- paramètres internes :
d : integer : compteur de caractères
i : integer : indice de boucle
lon : integer : longueur du string à décaler
conti : boolean : indique s'il faut continuer
usure : set of char : indique les caractères d'usure
medium : string : variable de travail
- fonction :
permettre de décaler dans le fichier résultat
un string de 'place' places.

- 152) - nom :
ligbla
- structure :
procédure
- mode d'appel :
ligbla
- besoin :
lig_res
- utilise :
écrire
- paramètre interne :
i : indice de boucle
- fonction :
écrire une ligne de blancs.
- 153) - nom :
num
- structure :
fonction
- mode d'appel :
num
- besoins :
lig_lu : string : idem
num : integer : le numéro indiqué dans la ligne
- utilise :
strint
- paramètres internes :
i : integer : indice de boucle
s : string : variable de travail
- fonction :
indiquer le numéro d'une variable indiquée dans la
ligne lig_lu.
- 154) - nom :
vardec
- structure :
procédure
- mode d'appel :
vardec
- besoin :
lig_lu : string : idem
- utilise :
liref24, num, lire
- paramètres internes :
dec : char : valeur de la variable de décision
val : boolean : valeur de la variable de décision
compas : boolean : correspondance des valeurs
stop : boolean : contrôle la fin de boucle
numero : integer : résultat de la fonction num
- fonction : gérer les variables de décisions.

155) - nom :
analy
- structure :
procedure
- mode d'appel :
analy
- besoin :
lig_lu : string : idem
- utilise :
num, vardec, info
- paramètre interne :
nur : integer : le numéro de la variable
- fonction :
déterminer s'il s'agit d'une variable de décision
ou d'une autre variable.

156) - nom :
resulb
- structure :
procedure
- mode d'appel :
resulb
- besoins :
sour : text : idem
lig_lu : string : idem
- utilise :
ecrire, ligbla, lire, analy
- fonction :
faire les lectures dans le fichier source jusqu'à
la fin de celui-ci.

157) - nom :
insf17
- structure :
procedure
- mode d'appel :
insf17
- besoins :
lig_lu : string : idem
lig_res : string : idem
- utilise :
liref17, decalage, liredescr
- paramètres internes :
trav : string : idem
trav2 : string : idem
trint : integer : numéro d'application
k : integer : nombre d'articles
i : integer : indice de boucle
lo : integer : nombre d'articles

d : integer : variable de travail
l : integer : variable de travail
nor : integer : variable de travail
term : boolean : fin fichier17
term1 : boolean : fin fichier description
- fonction :
insérer les applications et leur description.

158) - nom :
insf25
idem que la procédure insf17

159) - nom :
insf22
- structure :
procédure
- mode d'appel :
insf22
- besoin :
lig_res : string : idem
- utilise :
liref22, decalage, ligbla, tabl
- paramètres internes :
trav : string : idem
trav1 : string : idem
trav2 : string : idem
lo : integer : longueur de description
i : integer : indice de boucle
d : integer : nombre maximum d'articles
term : boolean : idem
tabla : tabla : idem
- fonction :
insérer les données du fichier 22.

161) - nom :
insf23
- structure :
procédure
- mode d'appel :
insf23 (nur)
- entrée :
nur : integer : numéro de la variable
- besoin :
lig_lu
- utilise :
strint, liref23, decalage

- paramètres internes :
 - trav : string : idem
 - trav2 : string : idem
 - a : string : nom étape
 - b : string : date étape
 - c : string : lieu étape
 - d : string : heure étape
 - term : boolean : idem
 - tabla : tab : informations à écrire
 - lo : integer : nombre d'articles
 - i : integer : indice de boucle
- fonction :
 - insérer les informations du fichier 23.

160) - nom :
insf21

- structure :
 - procedure
- mode d'appel :
 - insf21
- besoins :
 - lig_lu : string : idem
 - lig_res : string : idem
- utilise :
 - decalage, liref21 .
- paramètres internes :
 - trav : string : idem
 - trav1 : string : idem
 - trav2 : string : idem
 - term : boolean : idem
 - i : integer : indice de boucle
 - lo : integer : longueur de description
 - j : integer : indice de boucle
 - tabla : tab : idem
- fonction :
 - insérer les informations du fichier 21.

162) - nom :
insf1

- structure :
 - procedure
- mode d'appel :
 - insf1
- entrée :
 - nu : integer : numéro de la donnée
- utilise :
 - liref1, decalage
- paramètre interne :
 - trav : string : idem
- fonction :
 - insérer les informations du fichier 2.

163) - nom :
 insf2
- structure :
 procedure
- mode d'appel :
 insf2 (nu)
- entrée :
 nu : integer : idem
- utilise :
 tabla, decalage, liref2
- paramètres internes :
 trav : string : idem
 trint : integer : idem
 lo : integer : longueur de la description
 tabla : tab : la description
- fonction :
 insérer les informations du fichier 2.

164) - nom :
 insf6
- structure :
 procedure
- mode d'appel :
 insf6 (nu)
- entrée :
 nu : integer : idem
- besoin :
 lig_lu : string : idem
- utilise :
 decalage, liref6, tabl, lire
- paramètres internes :
 trav : string : idem
 trav2 : string : idem
 trav3 : string : idem
 term : boolean : idem
 tabla : tab : idem
 lo : integer : longueur de la description
- fonction :
 insérer les informations du fichier 6.

165) - nom :
 insf7
- structure :
 procedure
- mode d'appel :
 insf7 (nu)
- entrée :
 nu : integer : idem

- utilise :
liref7, decalage
- paramètres internes :
bb : integer : numéro de donnée
trint : integer : donnée lue
trav : string : donnée lue
trav2 : string : variable de travail
trav3 : string : variable de travail
- fonction :
insérer les données du fichier 7.

166) - nom :
insf11

- structure :
procedure
- mode d'appel :
insf11 (nu)
- entrée :
nu : integer :idem
- utilise :
liref11, decalage, soulig, tabl
- paramètres internes :
trav : string : nom d'application
term : boolean : idem
des : tab : description d'application
i : integer : commande l'ouverture ou la
fermeture du fichier 11
k : integer : longueur de description
j : integer : numéro de donnée
cpt : integer : indice de boucle
d : integer : nombre maximum d'applications
- fonction :
insérer les informations du fichier 11.

167) - nom :
insf12

- structure :
procedure
- mode d'appel :
insf12 (nu)
- entrée :
nu : integer : idem
- utilise :
liref12, decalage
- paramètre interne :
trav : string : information lue
- fonction :
insérer les informations du fichier 12.

171) - nom :
inter
- structure :
procedure
- mode d'appel :
inter
- besoins :
o_n : set of char : ensemble des réponses admises
rep : char : la réponse
- utilise :
lirechar, efecran
- fonction :
demande à l'écran si le jeu de caractères est compatible et si la longueur de la page est bonne.

172) - nom :
ouvert
- structure :
procedure
- mode d'appel :
ouvert (ok)
- sortie :
ok : boolean : indique si l'ouverture du fichier source s'est bien passée
- besoin :
nom : string : nom du fichier source
- utilise :
efecran, lirestring, openprinter, prcaractère
- paramètre interne :
i : integer : indice de boucle
- fonction :
ouvre le fichier source (maximum trois tentatives) et l'imprimante.

173) - nom :
fermer
- structure :
procedure
- mode d'appel :
fermer
- besoin :
impr : text : fenêtre sur le fichier source
- utilise :
efecran
- fonction :
ferme le fichier source et l'imprimante.

- 174) - nom :
pghead
- structure :
procedure
- mode d'appel :
pghead
- besoin :
cl : integer : compteur de ligne
- paramètre interne :
i : integer : indice de boucle
- fonction :
gène un entête de page.
- 175) - nom :
pgfoot
- structure :
procedure
- mode d'appel :
pgfoot
- besoin :
cp : integer : compteur de pages
- paramètres internes :
i : integer : indice de boucle
num : string : numéro de page
l : string : variable de travail
- fonction :
gène un bas de page (avec le n° de celle-ci).
- 176) - nom :
sautpg
- structure :
procedure
- mode d'appel :
sautpg
- besoins :
plt : integer : longueur totale d'une page
cl : integer : longueur de la page courante
- utilise :
pgfoot, pghead
- paramètres internes :
finpg : nombre de lignes pour atteindre la fin
de la page
i : integer : indice de boucle
- fonction :
générer un saut de page.

177) - nom :
trsf
- structure :
procédure
- mode d'appel :
trsf
- besoin :
cl : integer : compteur de lignes
- paramètres internes :
carprobl : set of char : ensemble de caractères à
problèmes
long : integer : longueur de ligne à écrire
i : integer : indice de boucle
c : char : le caractère problématique
- fonction :
transformer les caractères litigieux.

178) - nom :
impacc
- structure :
procédure
- mode d'appel :
impacc (nom_fich)
- entrée :
nom_fich : string : le nom du fichier à imprimer
- utilise :
inter, ouvert, pghead, sautpg, trsf, pgfoot
- paramètres internes :
trans : boolean : indique s'il faut transformer
ok : boolean : indique si tout est bien
ouvert
o_n : set of char : caractères de réponse
rep : char : réponse
plt : integer : longueur totale de la page
cp : integer : compteur de pages
cl : integer : compteur de lignes
i : integer : indice de boucle
pl : integer : longueur de page effective
fin : integer : fin de page
impr : text : fichier source
lig_ecr : string : ligne à écrire
nom : string : nom du fichier source
no : string : variable intermédiaire
rep1 : string : réponse
- fonction :
permet d'imprimer un fichier de nom donné.

179) - nom :
 exterieur
- structure :
 procedure
- mode d'appel :
 exterieur
- utilise :
 impacc, lirestring, efecran, lirechar
- paramètres internes :
 cont : boolean : arrête la boucle
 no : string : string lu à l'écran
 nom_fich : string : nom du fichier source
 rep1 : string : réponse
 o_n : set of char : ensemble de caractères
 admises
 rep : char : réponse
- fonction :
 permet d'imprimer un fichier en indiquant le nom
 de celui-ci à l'écran.

ANNEXE 12.11. : MODULE "GESTION ARBRE FOURNISSEURS"

- 180) - nom :
crea_arb_f
- structure :
procedure
- mode d'appel :
crea_arb_f (critere)
- entrée :
critere : integer : numéro de critère
- besoins :
heap, heap2, rac, rac2
- modifie :
rac, rac2, heap, heap2
- utilise :
mark, efecran, message, sel_fourn, arb_f,
verif_fourn
- paramètres internes :
nom_fourn : string : nom d'un fournisseur
num_rec_f : integer : numéro de record fournisseur
fin_sel : boolean : indique fin de sélection
ok_actif : boolean : indique la présence du
fournisseur
- fonction :
créé l'arbre des fournisseurs suivant le critère
de numéro "critere".
- 181) - nom :
det_arb_f
- structure :
procedure
- mode d'appel :
det_arb_f (critere)
- entrée :
critere : integer : numéro de critère
- besoins :
heap, heap2
- utilise :
release
- fonction :
détruit l'arbre des fournisseurs suivant le cri-
tère de numéro "critere".

182) - nom :
aj_arb_f
- structure :
procédure
- mode d'appel :
aj_arb_f (d_fourn, num_rec_f)
- entrées :
d_fourn : r8 : record fournisseur
num_rec_f : integer : numéro de record fournisseur
- besoin :
rac
- modifie :
rac
- utilise :
arb_f
- fonction :
ajoute à l'arbre des fournisseurs le fournisseur
de nom "d_fourn.nom" et de record numéro
"num_rec_f".

183) - nom :
sup_arb_f
- structure :
procédure
- mode d'appel :
sup_arb_f (arb, nom_f)
- entrée :
nom_f : string : nom d'un fournisseur
- entrée/sortie :
arb : pointalp : identificateur de l'arbre des
fournisseurs
- utilise :
sup_arb_f
- fonction :
supprime de l'arbre des fournisseurs le fournis-
seur de nom "nom_f".

184) - nom :
arb_f
- structure :
procédure
- mode d'appel :
arb_f (nom_f, num_rec_f, arb)
- entrées :
nom_f : string : nom d'un fournisseur
num_rec_f : integer : numéro de record fournisseur
- entrée/sortie :
arb : pointalp : identificateur de l'arbre des
fournisseurs

- utilise :
nerv, arb_f
 - fonction :
construit un élément de l'arbre "arb".
- 185)
- nom :
list_arb_f
 - structure :
procedure
 - mode d'appel :
list_arb_f (choix, arb)
 - entrée :
choix : char : caractère déterminant où lister
 - entrée/sortie :
arb : pointalp : identificateur de l'arbre de fournisseurs
 - utilise :
list_arb_f, lirenum, lister_f, imp_f
 - paramètre interne :
d_fourn : r8 : record fournisseur
 - fonction :
liste les informations fournisseur de l'arbre des fournisseurs "arb" à l'écran si choix : 'E' et à l'imprimante si choix = 'I'.
- 186)
- nom :
imp_f
 - structure :
procedure
 - mode d'appel :
imp_f (d_fourn)
 - entrée :
d_fourn : r8 : record fournisseur
 - utilise :
writeln, prpage, etat
 - paramètre interne :
etat2 : array [1..11] of char : état d'avancement listable
 - fonction :
imprime les informations fournisseurs à l'imprimante.
- 187)
- nom :
lister_f
 - structure :
procedure
 - mode d'appel :
lister_f (d_fourn)

- entrée :
 - d_fourn : r8 : record fournisseur
 - utilise :
 - efecran, gotoxy, write, lirereturn, etat
 - paramètre interne :
 - etat2 : array [1..11] of char : état d'avancement
affichable
 - fonction :
 - liste à l'écran les informations fournisseurs
- 188)
- nom :
 - etat
 - structure :
 - procedure
 - mode d'appel :
 - etat (etat1, etat2)
 - entrée :
 - etat1 : array [1..11] of integer : etats d'avancements
 - etat2 : array [1..11] of char : état d'avancement
listable
 - paramètre interne :
 - i : integer : variable de travail d'une boucle
 - fonction :
 - transforme "etat1" en "etat2" état d'avancement
affichable.
- 189)
- nom :
 - parc_arb_f
 - structure :
 - function : integer
 - mode d'appel :
 - parc_arb_f (nom_f)
 - entrée :
 - nom_f : string : nom d'un fournisseur
 - besoin :
 - rac
 - paramètre interne :
 - num_rec_f : integer : numéro de record fournisseur
 - fonction :
 - donne le numéro de record du fournisseur de
nom "nom_f"
- 190)
- nom :
 - parc_arb_num
 - structure :
 - procedure


```
- mode d'appel :  
  parc_arb_num (nom_f, arb, num_rec_f)  
- entrée :  
  nom_f : string      : nom d'un fournisseur  
  arb   : pointalp    : identificateur de l'arbre de  
                        fournisseurs  
- sortie :  
  num_rec_f : integer : numéro de record fournisseur  
- utilise :  
  parc_arb_num  
- fonction :  
  donne, depuis l'arbre de fournisseurs "arb", le  
  numéro de record "num_rec_f" du fournisseur de  
  nom "nom_f".
```

```
191) - nom :  
      list_sel_f  
- structure :  
  procedure  
- mode d'appel :  
  list_sel_f (x, arb, y)  
- entrée :  
  x   : integer      : coordonnée X  
  arb : pointalp    : coordonnée Y  
- entrée/sortie :  
  y   : integer      : coordonnée Y  
- utilise :  
  list_sel_f, gotoxy, write  
- fonction :  
  affiche à l'écran la liste des fournisseurs de  
  l'arbre "arb", et ce aligné à gauche sur la  
  coordonnée "X" et à partir de la "Y"ème ligne.
```

ANNEXE 12.12. : MODULE "SELECTION SUR CRITERE"

```
192) - nom :
      sel_fourn
- structure :
  procedure
- mode d'appel :
  sel_fourn (critere, nom_fourn, num_rec_f, fin_sel,
             ok_actif)
- entrée :
  critere : integer : nom du fournisseur sélectionné
- sorties :
  nom_fourn : string : nom du fournisseur sélectionné
  num_rec_f : integer : numéro de record du fournisseur
  fin_sel   : boolean : indique la fin de la sélection
  ok_actif  : boolean : indique la présence du fournisseur (i.e. un fournisseur
                    a bien été sélectionné)
- utilise :
  lirenumb
- paramètres internes :
  trouve    : boolean : indique qu'un fournisseur est trouvé
  switch_bd : boolean : indique la fin du fichier
  d_fourn   : r8      : record de fournisseur
- fonction :
  sélectionne un fournisseur de nom "nom_fourn", de
  numéro de record "num_rec_f", ce fournisseur
  répondant au critère de numéro "critere", de plus
  "fin_sel" indique la fin de la sélection et
  "ok_actif" si le fournisseur est présent.
- algorithme
  - trouve := false; fin_sel := false; ok_actif :=
    false;
  - tantque pas trouve et pas fin_sel faire
    - lire le record de numéro "num_rec_f" =: d_fourn;
    - ok_actif := d_fourn.actif;
    - si ok_actif (x le fournisseur est actif x)
      - alors - si critere = 0
        - alors - trouve := true
          - nom_fourn := d_fourn.nomf
            (x le fournisseur est d'office
            pris x)
```



```

|
|         sinon - si d_fourn.etat critere = 1
|                 alors trouve := true finsi;
|                 (x le fournisseur satisfait
|                 le critère x)
|                 - nom_fourn := d_fourn.nomf
|         finsi;
|         finsi;
| - fin_sel := switch_bd;
| - si not trouvé (x pas trouvé de fournisseur x)
|   alors num_rec_f := num_rec_f + 1
|   finsi;
| - si switch_bd et trouvé
|   alors ok_actif := false
|   finsi;
| finfaire;

```


ANNEXE 12.13. : MODULE "GESTION ECRAN"

- 193) - nom :
init_curseur
- structure :
procedure
- mode d'appel :
init_curseur (num_mat)
- entrée :
num_mat : integer : numéro de matrice
- besoin :
moteur, num_moteur
- modifie :
moteur, num_moteur
- utilise :
reset, rewrite, get, put, eof, seek, close,
ioresult
- paramètres internes :
i, j : integer : variables de travail de
boucle
fich_cur : f3 : fichier de matrice
- fonction :
place de la matrice, contenant les positions de
curseurs, de numéro "num_mat" en mémoire, si
elle n'existe pas, elle est initialisée à -1.
- 194) - nom :
curseur
- structure :
procedure
- mode d'appel :
curseur (num_mes, num_intro, num_mat)
- entrées :
num_mes : integer : numéro de message
num_intro : integer : numéro de l'introduction
num_mat : integer : numéro de la matrice con-
sidérée
- besoin :
moteur
- utilise :
init_curseur, gotoxy, dec_int
- paramètre interne :
x, y, x1, y1 : integer : coordonnées de curseur
- fonction :
positionne le curseur, dont les coordonnées appar-
tiennent à la matrice de numéro "num_mat", pour
le message numéro "num_mes" (= indice de colonne)
et pour l'introduction numéro "num_intro" (= in-
dice de ligne).

- 195) - nom :
 efecran
- structure :
 procedure
- mode d'appel :
 efecran
- utilise :
 gotoxy, write, chr
- fonction :
 nettoye l'écran.
- 196) - nom :
 test_goto
- structure :
 procedure
- mode d'appel :
 test_goto (x,y)
- entrée :
 x, y : integer : coordonnées de curseur
- utilise :
 erreur, exit
- fonction :
 si 1 (0 x 23 et 0 y 79) alors un message
 d'erreur est envoyé et on sort du programme.
- 197) - nom :
 ar_plan
- structure :
 procedure
- mode d'appel :
 ar_plan (f, x1, y1)
- entrée :
 f : threebits : numéro du ton désiré
 x1, y1 : integer : coordonnées du curseur
- utilise :
 test_goto, gotoxy, backgd
- fonction :
 définit l'arrière plan au ton f depuis les
 coordonnées (x1,y1).
- 198) - nom :
 pr_plan
- structure :
 procedure
- mode d'appel :
 pr_plan (f, x1, y1)

- entrées :
 - f : threebits : numéro du ton désiré
 - x1, y1 : integer : coordonnées du curseur
 - utilise :
 - test_goto, gotoxy, foregnd
 - fonction :
 - définit l'avant-plan au ton f depuis les coordonnées (x1,y1).
- 199) - nom :
clign_in
- structure :
 - procedure
 - mode d'appel :
 - clign_in (x1, y1)
 - entrée :
 - x1, y1 : integer : coordonnées du curseur
 - utilise :
 - test_goto, gotoxy, blink
 - fonction :
 - provoque le clignotement de l'écran à partir des coordonnées (x1,y1).
- 200) - nom :
clign_out
- structure :
 - procedure
 - mode d'appel :
 - clign_out (x1, y1)
 - entrée :
 - x1, y1 : integer : coordonnées du curseur
 - utilise :
 - test_goto, gotoxy, blink
 - fonction :
 - arrête le clignotement de l'écran à partir des coordonnées (x1,y1).
- 201) - nom :
message
- structure :
 - procedure
 - mode d'appel :
 - message (num_mes, num_mat, fich_mes)
 - entrées :
 - num_mes : integer : numéro de message
 - num_mat : integer : numéro de matrice
 - fich_mes : string : nom du fichier des écrans

- besoin :
- moteur
- utilise :
- gotoxy, recherche, envoie_message, init_curseur, dec_int
- paramètres internes :
- x, y : integer : coordonnées de curseur
- rech1, rech2 : integer : numéros de record dans fich_mes
- fonction :
- envoie à l'écran le message de numéro "num_mes", dont les coordonnées appartiennent à la matrice de numéro "num_mat", ce message appartenant au fichier fich_mes.

- 202) - nom :
- recherche
 - structure :
 - procedure
 - mode d'appel :
 - recherche (num_mes, fich_mes, rech1, rech2)
 - entrées :
 - num_mes : integer : numéro de message
 - fich_mes : string : nom du fichier des écrans
 - sorties :
 - rech1 : integer : numéro de record 1° ligne écran
 - rech2 : integer : numéro de record dernière ligne écran
 - utilise :
 - reset, ioresult, seek, get, close, erreur, exit
 - besoin :
 - anx_ecran : f2 : fichier annexe de pointeurs
 - paramètre interne :
 - fich_anx1 : string : nom fichier annexe de pointeurs
 - fonction :
 - recherche les numéros de record correspondant à la première et dernière ligne du message de numéro "num_mes" (du fichier "fich_mes").

- 203) - nom :
- envoie_message
 - structure :
 - procedure
 - mode d'appel :
 - envoie_message (rech1, rech2, fich_mes)
 - entrées :
 - rech1 : integer : numéro de record 1° ligne
 - rech2 : integer : numéro de record dernière ligne
 - fich_mes : string : nom du fichier des écrans

- utilise :
reset, ioresult, seek, get, writeln, close,
erreur, exit
- besoin :
fich_ecran : f1 : fichier des écrans
- fonction :
envoie à l'écran le message se trouvant entre
le rech1ème et le rech2ème record dans fich_mes.

- 204) - nom :
erreur
- structure :
procedure
 - mode d'appel :
erreur (s)
 - entrée :
s : string : message à envoyer
 - utilise :
gotoxy, blink, write, time, sc_erase_to_eol
 - fonction :
affiche à l'écran le string s, et ce aux coor-
données (32,23) pendant quelques secondes.

- 205) - nom :
texte_libre
- structure :
procedure
 - mode d'appel :
texte_libre (okset, endroit, s, d_endroit)
 - entrées :
okset : set of char : ensemble des caractères
admissibles
endroit : integer : numéro de ligne (début du
texte)
 - sortie :
s : string : string lu pour une ligne
 - entrée/sortie :
d_endroit : integer : numéro de ligne
 - utilise :
gotoxy, lirestring, sc_eras_eos, write
 - fonction :
permet de lire un string "s", dont les caractères
appartiennent à okset, commençant à la ligne
numéro "d_endroit", ceci faisant partie de la
gestion d'un texte libre commençant à la ligne
"endroit".

206) - nom :
lirechar
- structure :
procédure
- mode d'appel :
lirechar (okset, carlu)
- entrée :
okset : set of char : ensemble de caractères
admissibles

- sortie :
carlu : char : caractère lu
- utilise :
lirestring
- paramètres internes :
longueur : integer : longueur du caractère (= 1 !)
stringlu : string : string de longueur 1 lu
- fonction :
lecture d'un caractère "carlu" appartenant à
l'ensemble "okset".

207) - nom :
lirestring
- structure :
procédure
- mode d'appel :
lirestring (okset, longueur, s)
- entrées :
okset : set of char : ensemble de caractères
admissibles
longueur : integer : longueur maximale du
string

- sortie :
s : string : string lu à la console
- besoins :
cr, bs
- utilise :
sc_find_x, sc_find_y, gotoxy, length, lirecar,
concat, delete, write
- paramètres internes :
long : integer : longueur "légale" du string
l : integer : longueur du string (et ses
versions)
x, y : integer : coordonnées de curseur
i : integer : variable de travail d'une
boucle
S1 : string : string lu (et ses versions)
car : string 1 : un caractère lu
lireset : set of char : ensemble des caractères
admissibles y compris cr et
bs

- fonction :
lecture à la console d'un string "s" de longueur maximale "longueur" et dont tous les caractères sont dans l'ensemble "okset".
- commentaire :
l'algorithme de base est donné par H. Haut [7].

- 208)
- nom :
lirecar
 - structure :
fonction : char
 - mode d'appel :
lirecar (okset)
 - entrée :
okset : set of char : ensemble des caractères admissibles
 - besoins :
bell, bs, ensacc
 - utilise :
read, eoln, write
 - paramètres internes :
ok : boolean : = vrai si ch dans okset
ch : char : 1 ! caractère lu
 - fonction :
donne la valeur d'un caractère lu à la console et appartenant à l'ensemble "okset".
 - commentaire :
l'algorithme de base est donné par H. Haut [7].

- 209)
- nom :
lirereturn
 - structure :
procédure
 - mode d'appel :
lirereturn
 - besoin :
cr
 - utilise :
lirechar
 - paramètres internes :
ch : char : caractère lu par lirechar
okset: set of char : ensemble des caractères admissibles
 - fonction :
attent que la touche RETURN soit pressée.

210) - nom :
lireentier
- structure :
 fonction : integer
- mode d'appel :
 lireentier
- besoins :
 numset, signeset, cr, bs, bell
- utilise :
 ord, length, lirecar, trunc, abs, div, write,
 concat, strtoint, delete
- paramètres internes :
 longueur : integer : longueur du string lu
 valeur : integer : valeur de l'entier
 z : integer : variable de travail
 maxint : integer : valeur maximale d'un entier
 s : string : string lu
 car : string 1 : un caractère lu
 okset : set of char : ensemble des caractères
 admissibles
 lireset : set of char : ensemble des caractères
 admissibles y compris cr et bs
- fonction :
 retourne la valeur d'un entier lu à la console
 (il ne sera pas supérieur à 32767).
- commentaire :
 l'algorithme de base est donné par H. Haut [7].

210b) - nom :
lirecar2
- structure :
 fonction : char
- mode d'appel :
 lirecar2 (okset)
- entrée :
 okset : set of char : ensemble des caractères
 admissibles
- besoin :
 bell
- utilise :
 read, eolu, chr, write
- paramètres internes :
 ok : boolean : vrai = caractère lu est dans okset
 ch : char : un caractère lu
- fonction :
 lecture d'un caractère, sur la keyboard, appar-
 tenant à "okset"

210c) - nom :
lirecode
- structure :
procédure
- mode d'appel :
lirecode (okset, S)
- entrée :
okset : set of char : ensemble des caractères
admissibles

- besoins :
cr, bs
- utilise :
length, lirecar2, concat, delete
- paramètres internes :
l : integer : longueur du code lu
long : integer : longueur maximale d'un code
S1 : string : code lu (et ses différentes
versions)
car : string 1 : un caractère lu
lireset : set of char : ensemble des caractères
admissibles, y compris cr et bs

- fonction :
lecture d'un mot de passe (sur la keyboard) dont
les caractères appartiennent à "okset".

211) - nom :
lirecom
- structure :
procédure
- mode d'appel :
lirecom (coms, l)
- sorties :
coms : tab3 : trois lignes (maximum) lues
l : integer : nombre de lignes
- besoin :
enscar
- utilise :
efecran, message, curseur, sc_find_x, sc_find_y,
gotoxy, lirestring
- paramètres internes :
x, y : integer : coordonnées de curseur
comp : integer : indice de coms
i : integer : variable de boucle
s : string : chaîne de caractères lue à
l'écran
ens_texte : set of char : ensemble de caractères
admissibles

- fonction :
lit à l'écran au maximum 3 lignes contenues dans
"coms", le nombre de lignes est contenu dans l.

212) - nom :
ecriecran
- structure :
procEDURE
- mode d'appel :
ecriecran (long, tableau)
- entrées :
long : integer : longueur du tableau à écrire
tableau : tab : tableau de chaînes de caractères à écrire à l'écran
- utilise :
writeln
- paramètre interne :
i : integer : variable de boucle
- fonction :
affiche à l'écran le tableau "tableau" de chaînes de caractères et de longueur "long".

213) - nom :
lirecodage
- structure :
procEDURE
- mode d'appel :
lirecodage (rep)
- sortie :
rep : string : codage lu à l'écran
(i.e. O, o, N, n, O*, o*, N*, n*, *, -)
- besoins :
cr, bs
- utilise :
lirecar, copy, concat
- paramètres internes :
okset : set of char : ensemble de caractères admissibles
C1, C2, C3 : char : caractères lus à l'écran
S1, S2, S3, S4 : string : chaînes de caractères de travail
- fonction :
lecture à l'écran du codage.

ANNEXE 12.14. : MODULE "GESTION IMPRIMANTE"

(d'après H. Haut [7])

214) - nom :
openprinter
- structure :
procédure
- mode d'appel :
openprinter
- utilise :
close, reset
- fonction :
définit l'imprimante comme output en vigueur.

215) - nom :
closeprinter
- structure :
procédure
- mode d'appel :
closeprinter
- utilise :
close, reset
- fonction :
redéfinit la console comme output en vigueur.

216) - nom :
prpage
- structure :
procédure
- mode d'appel :
prpage
- utilise :
page
- fonction :
provoque un saut de page à l'imprimante.

217) - nom :
prsaute ligne
- structure :
procédure
- mode d'appel :
prsaute ligne (I)

- entrée :
I : integer : nombre de lignes feed à envoyer
- utilise :
writeln
- paramètre interne :
k : integer : variable de travail pour une boucle
- fonction :
envoie "i" ligne feed à l'imprimante

- 218) - nom :
prfindepage
- structure :
procedure
 - mode d'appel :
prfindepage (i)
 - entrée :
i : integer : nombre de lignes à sauter en bas
de page
 - besoin :
 - so
 - utilise :
write, chr
 - fonction :
définit le nombre de lignes "i" à sauter en bas
de page.

- 219) - nom :
prlongueur
- structure :
procedure
 - mode d'appel :
prlongueur (i)
 - entrée :
i : integer : longueur de page en nombre de lignes
 - besoin :
 - fl
 - utilise :
write, chr
 - fonction :
définit une longueur de page de "i" lignes.

- 220) - nom :
prcaractères
- structure :
procedure
 - mode d'appel :
prcaractères (i)

- entrée :
i : integer : numéro de set de caractères
- besoin :
cs
- utilise :
write, chr
- fonction :
définit le set de caractères à utiliser.

- 221) - nom :
prlargeur
- structure :
procedure
 - mode d'appel :
prlargeur (i)
 - entrée :
i : integer : largeur de page en nombre de
caractères
 - besoin :
crv
 - utilise :
write, chr
 - fonction :
définit une largeur de page de "i" caractères

- 222) - nom :
prcondense
- structure :
procedure
 - mode d'appel :
prcondense
 - besoin :
co
 - utilise :
write
 - fonction :
met l'imprimante en mode condensé.

- 223) - nom :
prelargi
- structure :
procedure
 - mode d'appel :
prelargi
 - besoin :
el
 - utilise :
write
 - fonction :
met l'imprimante en mode élargi.

224) - nom :
praccentue
- structure :
procedure
- mode d'appel :
praccentue
- besoin :
em
- utilise :
write
- fonction :
met l'imprimante en mode accentué.

225) - nom :
prdouble
- structure :
procedure
- mode d'appel :
prdouble
- besoin :
dO
- utilise :
write
- fonction :
met l'imprimante en mode double.

226) - nom :
prnormal
- structure :
procedure
- mode d'appel :
prnormal
- besoin :
no
- utilise :
write
- fonction :
met l'imprimante en mode normal.

ANNEXE 12.15. : MODULE "BASE DE DONNEES"

227) - nom :
if1
- structure :
procedure
- mode d'appel :
if1
- besoins :
fI1.data
- modifie :
fI1.data
- paramètre interne :
f1a : fichier 1 (fenêtre)
- fonction :
initialise le fichier fI1.data

228) - nom :
if2
- structure :
procedure
- mode d'appel :
if2
- besoins :
fI2.data
- modifie :
fI2.data
- paramètre interne :
f2a : fichier 2 (fenêtre)
- fonction :
initialise le fichier fI2.data

229) - nom :
if7
- structure :
procedure
- mode d'appel :
if7
- besoins :
fI7.data
- modifie :
fI7.data
- paramètre interne :
f7a : fichier 7 (fenêtre)
- fonction :
initialise le fichier fI7.data

- 230) - nom :
if12
- structure :
procedure
- mode d'appel :
if12
- besoins :
fI12.data
- modifie :
fI12.data
- paramètre interne :
f12a : fichier 12 (fenêtre)
- fonction :
initialise le fichier fI12.data
- 231) - nom :
if13
- structure :
procedure
- mode d'appel :
if13
- besoins :
fI13.data
- modifie :
fI13.data
- paramètre interne :
f13a : fichier 13 (fenêtre)
- fonction :
initialise le fichier fI13.data
- 232) - nom :
if16
- structure :
procedure
- mode d'appel :
if16
- besoins :
fI16.data
- modifie :
fI16.data
- paramètres internes :
f16a : fichier 16 (fenêtre)
s : string : permet d'initialiser une valeur
i : integer : permet de gérer une boucle
- fonction :
initialise le fichier fI16.data

- 233) - nom :
if17
- structure :
procedure
- mode d'appel :
if17
- besoins :
fI17.data
- modifie :
fI17.data
- paramètres internes :
i : integer : permet de gérer une boucle
f17a : fichier 17 (fenêtre)
- fonction :
initialise le fichier 17.
- 234) - nom :
if21
- structure :
procedure
- mode d'appel :
if21
- besoins :
fI21.data
- modifie :
fI21.data
- paramètres internes :
i, j : integer : permet de gérer deux boucles
f21a : fichier 21 (fenêtre)
- fonction :
initialise le fichier 21.
- 235) - nom :
if22
- structure :
procedure
- mode d'appel :
if22
- besoins :
fI22.data
- modifie :
fI22.data
- paramètres internes :
i, j : integer : permet de gérer deux boucles
f22a : fichier 22 (fenêtre)
- fonction :
initialise le fichier 22.

- 236) - nom :
if23
 - structure :
procedure
 - mode d'appel :
if23
 - besoin :
fI23.data
 - modifie :
fI23.data
 - paramètres internes :
i : integer : permet de gérer une boucle
f23a : fichier 23 (fenêtre)
 - fonction :
initialise le fichier 23.
-
- 237) - nom :
if24
 - structure :
procedure
 - mode d'appel :
if24
 - besoin :
fI24.data
 - modifie :
fI24.data
 - paramètres internes :
i : integer : permet de gérer une boucle
f24a : fichier 24 (fenêtre)
 - fonction :
initialise le fichier 24.
-
- 238) - nom :
if25
 - structure :
procedure
 - mode d'appel :
if25
 - besoin :
fI25.data
 - modifie :
fI25.data
 - paramètres internes :
i, j : integer : permet de gérer deux boucles
f25a : fichier 25 (fenêtre)
 - fonction :
initialise le fichier 25.

- 239) - nom :
if26
- structure :
procedure
- mode d'appel :
if26
- besoin :
fI26.data
- modifie :
fI26.data
- paramètres internes :
i : integer : permet de gérer une boucle
f26a : fichier 26 (fenêtre)
- fonction :
initialise le fichier 26.
- 240) - nom :
patience
- structure :
procedure
- mode d'appel :
patience
- utilise :
efface
- fonction :
écrit à l'écran un message.
- 240b) - nom :
efface
- structure :
procedure
- mode d'appel :
efface
- fonction :
efface l'écran.
- 241) - nom :
init
- structure :
procedure
- mode d'appel :
init
- paramètre interne :
ndisk : string : indique le drive sur lequel il
faudra travailler
- utilise :
if1, if2, if7, if12, if13, if16, if17, if21, if22,
if23, if24, if25, if26, patience, efface
- fonction :
déclenche l'initialisation des fichiers de la
B.D.A.

- 242) - nom :
choice
- structure :
procedure
- mode d'appel :
choice
- besoins :
fI8.data : fichier des fournisseurs
fichier : string : nom du fichier
- fonction :
ouvre le fichier des fournisseurs sur =S s'il
existe, sur =4 sinon.
- 243) - nom :
verif_fourn
- structure :
function
- mode d'appel :
verif_fourn (nom) : boolean
- entrée :
nom : string : le nom d'un fichier
- sortie :
verif_fourn : boolean : indique si le fichier
existe
- besoins :
fI8.data : file of r8 : le fichier des fournisseurs
- fonction :
vérifie si le fichier (nom) existe.
- 244) - nom :
ecrfourn
- structure :
procedure
- mode d'appel :
ecrfourn (nom, fourni, ci, nfourn)
- entrées :
nom : string : nom du fichier
fourni : r8 : l'article fournisseur à écrire
ci : integer : indicateur d'ouverture/ferme-
ture du fichier
- sortie :
nfourn : integer : indique le numéro d'article
courant
- besoins :
fI8.data : r8 : fichier des fournisseurs
nfourn2 : integer : compteur de fournisseurs
- modifie :
fI8.data
nfourn2
- fonction :
écrit un fournisseur dans le fichier et retourne
le numéro d'article.

- 245) - nom :
modfourn
- structure :
procedure
- mode d'appel :
modfourn (fourn, numfourn, ct)
- entrées :
fourn : r8 : un article fournisseur
numfourn : integer: numéro du fournisseur
ct : boolean: indiquant si c'est la
dernière modification
- besoin :
fI8.data
- modifie :
fI8.data
- fonction :
modifie l'article fournisseur numéro numfourn.
- 246) - nom :
lirenumb
- structure :
procedure
- mode d'appel :
lirenumb (num, fournlu, finfich)
- entrée :
num : integer : numéro d'un fournisseur
- sorties :
fournlu : r8 : article fournisseur
finfich : boolean : indication de fin de fichier
- besoin :
fI8.data
- paramètre interne :
n : integer : compteur d'articles
- fonction :
permet de lire un article fournisseur de numéro
"num" et indique si on a lu le dernier article
ou pas.
- 247) - nom :
initetat
- structure :
procedure
- mode d'appel :
initetat
- besoin :
fI8.data
- modifie :
fI8.data

- paramètres internes :
 - n : integer : nombre de fournisseurs
 - j : integer : indice de boucle
 - i : integer : indice de boucle
- fonction :
 - permet d'initialiser tous les états d'avancements.

- 248) - nom :
ecrnum
- structure :
procedure
 - mode d'appel :
ecrnum (nom, fourni, num)
 - entrées :
 - num : integer : numéro du fournisseur à écrire
 - nom : string : nom du fichier
 - fourni : r8 : fournisseur à écrire
 - besoin :
fI8.data
 - modifie :
fI8.data
 - fonction :
permet d'écrire le fournisseur "num" dans le fichier "nom".

- 249) - nom :
lirefourn
- structure :
procedure
 - mode d'appel :
lirefourn (dr, ci, fournlu, f8fin)
 - entrées :
 - dr : string : nom du fichier
 - ci : integer : indication d'ouverture/fermeture de fichier
 - sorties :
 - fournlu : r8 : fournisseur lu
 - f8fin : boolean : indicateur de fin de fichier
 - besoins :
fI8.data
 - nfou1 : integer : compteur d'articles
 - nfour : integer : indicateur du nombre d'articles
 - paramètre interne :
 - terminer : boolean : indique si la fin du fichier a été atteint
 - fonction :
permet de lire séquentiellement le fichier des fournisseurs, et à la fin ferme le fichier.

250) - nom :
lirebfourn
idem que lirefourn, sauf que le fichier n'est pas fermé d'office à la fin du fichier.

250b) - nom :
lirenum
- structure :
procédure
- mode d'appel :
lirenum (num, fournlu)
- entrée :
num : integer : le numéro du fournisseur à lire
- sortie :
fournlu : r8 : le fournisseur lu
- besoin :
fI8.data
- fonction :
lire un fournisseur de numéro "num" et refermer le fichier.

251) - nom :
chng_disk
- structure :
procédure
- mode d'appel :
chng_disk (old, new)
- entrées :
old : string : le nom de la disquette à changer
new : string : nouveau nom de la disquette
- besoin :
disquette de nom old
- modifie :
le nom de la disquette
- paramètre interne :
rslt : d_result : cfr. appel système
- fonction :
permet de changer le nom d'une disquette.

252) - nom :
strint
- structure :
fonction
idem que STRTOINT dans le module technique.

- 252b) - nom :
 erbis
- structure :
 procedure
idem que ERREUR dans le module technique.
- 253) - nom :
 ecritt1
- structure :
 procedure
- mode d'appel :
 ecritt1 (nvar, valeur, ct)
- entrées :
 nvar : integer : numéro de la donnée
 valeur : string : valeur de la donnée
 ct : boolean : indicateur de fin de texte
- besoins :
 f01 : boolean : indicateur de fichier ouvert
 fI9.data, fI10.data, fI14.data, fI15.data,
 fI20.data, fI27.data : file of string :
 fichier de texte libre
 ft1 : file of string : fenêtre du fichier courant
- modifie :
 fI9.data, fI10.data, fI14.data, fI15.data,
 fI20.data, fI27.data
- fonction :
 écrire les textes libres de longueur quelconque
 et numéro "nvar".
- 254) - nom :
 ecritf1
- structure :
 procedure
- mode d'appel :
 ecritf1 (nvar, valeur)
- entrées :
 nvar : integer : idem
 valeur : string : idem
- besoin :
 fI1.data : le fichier de données 1
- modifie :
 fI1.data
- paramètre interne :
 f1a : file of r1 : fenêtre du fichier 1
- fonction :
 permet d'écrire les informations nvar dans le
 fichier 1.

255) - nom :
 ecritf2
- structure :
 procedure
- mode d'appel :
 ecritf2 (nvar, valeur, valint, ct)
- entrées :
 nvar : integer : idem
 valeur : string : idem
 valint : integer : un entier à écrire dans un
 fichier
 ct : boolean : idem
- besoins :
 fI2.data : file of r2
 f2a : file of r2 : fenêtre du fichier 2
- modifie :
 fI2.data
- paramètres internes :
 i : integer : indice de longueur de description
 f : boolean : indique s'il faut fermer le fichier
- fonction :
 permet d'écrire les informations nvar dans le
 fichier 2.

257) - nom :
 art
- structure :
 procedure
- mode d'appel :
 art (valeur, ct)
- entrées :
 valeur : string : idem
 ct : boolean : idem
- besoins :
 fI4.data
 f4a : file of r4 : fenêtre du fichier 4
 cptquest : compteur d'articles
- modifie :
 fI4.data, cptquest
- paramètre interne :
 i : integer : indice de longueur de question
- fonction :
 permet d'écrire les différentes questions.

258) - nom :
 ecritf4
- structure :
 procedure
- mode d'appel :
 ecritf4 (valeur, ct, ci)

- entrées :
 - valeur : string : idem
 - ct : boolean : idem
 - ci : integer : idem
- besoins :
 - fI4.data
 - f4a : file of r4 : fenêtre du fichier 4
- utilise :
 - art (ci-dessus)
- paramètre interne :
 - i : integer : indice de longueur de description
- fonction :
 - commande l'ouverture, l'écriture et la fermeture du fichier 4.

- 259) - nom :
art
- structure :
 - procedure
 - mode d'appel :
 - art (nvar, valeur)
 - entrées :
 - nvar : integer : idem
 - valeur : string : idem
 - besoins :
 - fI6.data
 - f6a : file of r6 : fenêtre du fichier 6
 - modifie :
 - fI6.data
 - fonction :
 - écrit dans le fichier 6 les données nvar.

- 260) - nom :
ecritf6
- structure :
 - procedure
 - mode d'appel :
 - ecritf6 (nvar, valeur, ci)
 - entrées :
 - nvar : integer : idem
 - valeur : string : idem
 - ci : integer : idem
 - besoins :
 - fI6.data
 - f6a : file of r6 : fenêtre du fichier 6
 - modifie :
 - fI6.data
 - utilise :
 - art
 - fonction :
 - commande l'ouverture, l'écriture et la fermeture du fichier 6.

- 261) - nom :
 ecritf7
- structure :
 procedure
- mode d'appel :
 ecritf7 (nvar, valeur, valint)
- entrées :
 nvar : integer : idem
 valeur : string : idem
 valint : integer : idem
- besoins :
 fI7.data
- modifie :
 fI7.data
- paramètre interne :
 f7a : file of r7 : fenêtre du fichier 7
- fonction :
 permet d'écrire dans le fichier 7.
- 262) - nom :
 art
- structure :
 procedure
- mode d'appel :
 art (nvar, valeur, ct)
- entrées :
 nvar : integer : idem
 valeur : string : idem
 ct : boolean : idem
- besoins :
 fI11.data
 f11a : file of r11 : fenêtre du fichier 11
- modifie :
 fI11.data
- paramètre interne :
 i : integer : indice de longueur de description
- fonction :
 permet d'écrire les articles du fichier 11.
- 263) - nom :
 ecritf11
- structure :
 procedure
- mode d'appel :
 ecritf11 (nvar, valeur, ct, ci)
- entrées :
 nvar : integer : idem
 valeur : string : idem
 ct : boolean : idem
 ci : integer : idem

- besoins :
 - fI11.data
 - f11a : file of r11 : fenêtre du fichier 11
 - utilise :
 - art
 - fonction :
 - gère l'ouverture, l'écriture et la fermeture du fichier 11.
- 264)
- nom :
 - ecritf12
 - structure :
 - procedure
 - mode d'appel :
 - ecritf12 (nvar, valeur)
 - entrées :
 - nvar : integer : idem
 - valeur : string : idem
 - besoin :
 - fI12.data
 - modifie :
 - fI12.data
 - paramètre interne :
 - f12a : file of r12 : fenêtre du fichier 12
 - fonction :
 - permet d'écrire dans le fichier 12.
- 265)
- nom :
 - ecritf13
 - structure :
 - procedure
 - mode d'appel :
 - ecritf13 (nvar, valeur)
 - entrées :
 - nvar : integer : idem
 - valeur : string : idem
 - besoin :
 - fI13.data
 - modifie :
 - fI13.data
 - paramètre interne :
 - f13a : file of r13 : fenêtre du fichier 13
 - fonction :
 - permet d'écrire dans le fichier 13.

269) - nom :
 ecritf21
- structure :
 procedure
- mode d'appel :
 ecritf21 (nvar, valeur, valint, ct)
- entrées :
 nvar : integer : idem
 valeur : string : idem
 valint : integer : idem
 ct : boolean : idem
- besoins :
 fI21.data
 f21a : file of r21 : fenêtre du fichier 21
- modifie :
 fI21.data
- paramètres internes :
 i : integer : compteur d'articles
 j : integer : indicateur de longueur de description
- fonction :
 écrire dans le fichier 21.

270) - nom :
 ecritf22
- structure :
 procedure
- mode d'appel :
 ecritf22 (nvar, valeur, valint, ct)
- entrées :
 nvar : integer : idem
 valeur : string : idem
 valint : integer : idem
 ct : boolean : idem
- besoins :
 fI22.data
 f22a : file of r22 : fenêtre du fichier 22
- modifie :
 fI22.data
- paramètres internes :
 i : integer : compteur d'articles
 j : integer : indicateur de longueur de description
- fonction :
 écrire dans le fichier 22.

271) - nom :
 ecritf23
- structure :
 procedure
- mode d'appel :
 ecritf23 (nvar, valeur, ct)

- entrées :
 - nvar : integer : idem
 - valeur : string : idem
 - ct : boolean : idem
- besoins :
 - fI23.data
 - f23a : file of r23 : fenêtre du fichier 23
- modifie :
 - fI23.data
- paramètre interne :
 - i : integer : compteur d'articles
- fonction :
 - écrire dans le fichier 23.

272) - nom :
ecrloca

- structure :
 - procedure
- mode d'appel :
 - ecrloca (numapl, valeur, ct)
- entrées :
 - numapl : integer : numéro de l'application
 - valeur : string : idem
 - ct : boolean : idem
- paramètres internes :
 - nfich : string : nom du fichier
 - s : string : nom du fichier
- besoin :
 - fo3 : boolean : indique si un fichier est ouvert
- fonction :
 - écrire la description d'un service.

273) - nom :
ecrdescr
idem que ecrloca

274) - nom :
ecritf24

- structure :
 - procedure
- mode d'appel :
 - ecritf24 (nvar, ct)
- entrées :
 - nvar : integer : idem
 - ct : boolean : variable de décision
- besoin :
 - fI24.data

- modifie :
fI24.data
- paramètre interne :
f24a : file of array of boolean : fenêtre du fichier 24
- fonction :
écrire dans le fichier 24.

- 275)
- nom :
ecrit
 - structure :
procedure
 - mode d'appel :
ecrit (nvar, valeur, valint, ct, ci, numappl)
 - entrées :
nvar : integer : idem
valeur : string : idem
valint : integer : idem
ct : boolean : idem
ci : integer : idem
 - sortie :
numappl : integer : numéro d'application
 - utilise :
ecritf1, escritf2, escritf4, escritf6, escritf7,
ecritf11, escritf12, escritf13, escritf16, escritf17,
ecrittl, escritf21, escritf22, escritf23, escritf24,
ecritf25
 - paramètre interne :
nvarbis : integer : la partie entière de nvar
divisé par 100
 - fonction :
réalise l'aiguillage vers les différentes procédures d'écriture.

- 276)
- nom :
nrappl
 - structure :
function
 - mode d'appel :
nrappl (noma)
 - entrée :
noma : string : le nom d'une application
 - sortie :
nrappl : integer : le numéro d'une application
 - besoin :
fI17.data
 - paramètres internes :
contenue : boolean : conducteur de boucle
f17a : file of r17 : fenêtre du fichier 17
 - fonction :
retourne le numéro de l'application dont a été
indiqué le nom.

277) - nom :
ecrf11b
- structure :
procedure
- mode d'appel :
ecrf11b (valeur, ct)
- entrées :
valeur : string : idem
ct : boolean : idem
- besoins :
fI11.data
f11a : file of r11 : fenêtre du fichier 11
- modifie :
fI11.data
- paramètre interne :
j : integer : indicateur de longueur de description
- fonction :
écrire dans le fichier 11 des descriptions unique-
ment.

278) - nom :
ecrf26
- structure :
procedure
- mode d'appel :
ecrf26 (nvar, cara)
- entrées :
nvar : integer : idem
cara : char : caractère à écrire
- besoin :
fI26.data
- modifie :
fI26.data
- paramètres internes :
i : integer : guide
f26a : file of array of char : fenêtre du fi-
chier 26
- fonction :
écrire dans le fichier 26.

279) - nom :
liref1
- structure :
procedure
- mode d'appel :
liref1 (nvar, valeurlu)
- entrée :
nvar : integer : idem

- sortie :
valeurlu : string : la donnée lue
- besoin :
fI1.data
- paramètre interne :
f1a : file of r1 : fichier de données 1
- fonction :
lire le fichier 1.

280) - nom :
liref2

- structure :
procedure
- mode d'appel :
liref2 (nvar, valeurlu, valintlu, long, tableaulu)
- entrée :
nvar : integer : idem
- sorties :
valeurlu : string : idem
valintlu : integer : entier lu
long : integer : la longueur de la description
tableaulu : array of string : une description lue
- besoins :
fI2.data
f2a : file of r2 : fenêtre du fichier 2
- paramètre interne :
i : integer : indice de boucle
- fonction :
lire le fichier 2.

281) - nom :
artlu

- structure :
procedure
- mode d'appel :
artlu
- besoins :
cptquest : integer : compteur de questions
fI4.data
f4a : file of r4 : fenêtre du fichier 4
- paramètre interne :
j : integer : indice de boucle
- fonction :
lire une question.

282) - nom :
liref4
- structure :
procédure
- mode d'appel :
liref4 (ci, valintlu, long, tableaulu, finfich)
- entrée :
ci : integer : idem
- sorties :
valintlu : integer : idem
long : integer : idem
tableaulu : tab : idem
finfich : boolean : indicateur de fin de fichier
- besoins :
cptquest : integer : compteur de questions
fI4.data
f4a : file of r4 : fenêtre du fichier 4
- utilise :
artlu
- paramètre interne :
f : boolean : indique la fermeture du fichier
- fonction :
lire le fichier 4.

283) - nom :
liref7
- structure :
procédure
- mode d'appel :
liref7 (nvar, valintlu, valeurlu)
- entrée :
nvar : integer : idem
- sorties :
valintlu : integer : idem
valeurlu : string : idem
- besoin :
fI7.data
- paramètre interne :
f7a : file of r7 : fenêtre du fichier 7
- fonction :
lire le fichier 7.

284) - nom :
artlu
- structure :
procédure
- mode d'appel :
artlu

- besoins :
 - fI6.data
 - f6a : file of r6 : fenêtre du fichier 6
 - modifiée :
 - f6a
 - paramètre interne :
 - j : integer : indice de boucle
 - fonction :
 - lire un article du fichier 6.
- 285)
- nom :
 - liref6
 - structure :
 - procedure
 - mode d'appel :
 - liref6 (ci, valeurlu, normeslu2, composlu3, finfich)
 - entrée :
 - ci : integer : idem
 - sorties :
 - valeurlu : string : idem
 - normeslu2 : string : information lue
 - composlu3 : string : information lue
 - finfich : boolean : idem
 - besoins :
 - fI6.data
 - f6a : fenêtre du fichier 6
 - utilise :
 - artlu
 - paramètre interne :
 - f : boolean : idem
 - fonction :
 - lire le fichier 6.
- 286)
- nom :
 - artlu
 - structure :
 - procedure
 - mode d'appel :
 - artlu
 - besoins :
 - fI11.data
 - f11a : fenêtre du fichier 11
 - paramètre interne :
 - j : integer : indice de boucle
 - fonction :
 - lire un article du fichier 11.

287) - nom :
liref11
- structure :
procédure
- mode d'appel :
liref11 (nvar, ci, long, valeurlu, valintlu,
finfich)
- entrées :
nvar : integer : idem
ci : integer : idem
- sorties :
long : integer : idem
valeurlu : string : idem
tableaulu : tab : idem
finfich : boolean : idem
- besoins :
i11 : integer : compteur d'articles
fI11.data
f11a : file of r11 : fenêtre du fichier 11
- utilise :
artlu
- paramètre interne :
f : boolean : idem
- fonction :
lire le fichier 11.

288) - nom :
liref11b
- structure :
procédure
- mode d'appel :
liref11b (ci, valeurlu, finfich)
- entrée :
ci : integer
- sorties :
valeurlu : string : idem
finfich : boolean : idem
- besoins :
cptf11 : integer : compteur
fI11.data
f11a : file of r11 : fenêtre du fichier 11
finfi11 : boolean : indicateur de fin de fichier
- fonction :
lire le nom d'une application à long terme.

289) - nom :
liref12
- structure :
procédure
- mode d'appel :
liref12 (nvar, valeurlu)

- entrée :
 - nvar : integer : idem
- sortie :
 - valeurlu : string : idem
- besoin :
 - fI12.data
- paramètre interne :
 - f12a : file of r12 : fenêtre du fichier 12
- fonction :
 - lire le fichier 12.

290) - nom :

- liref13

- structure :
 - procedure
- mode d'appel :
 - liref13 (nvar, valeurlu)
- entrée :
 - nvar : integer : idem
- sortie :
 - valeurlu : string : idem
- besoin :
 - fI13.data
- paramètres internes :
 - i : integer : guide
 - f13a : file of r13 : fenêtre du fichier 13
- fonction :
 - lire le fichier 13.

291) - nom :

- liref16

- structure :
 - procedure
- mode d'appel :
 - liref16 (ci, long, valeurlu, finfich)
- entrée :
 - ci : integer : idem
- sorties :
 - long : integer : indicateur du nombre d'articles
 - valeurlu : string : idem
 - finfich : boolean : idem
- besoins :
 - fI16.data
 - f16a : file of string : fenêtre du fichier 16
- paramètres internes :
 - i : integer : indicateur du nombre d'articles
 - f : boolean : idem
- fonction :
 - lire fichier 16.

292) - nom :
liref17
- structure :
procédure
- mode d'appel :
liref17 (ci, long, num, nom, finfich)
- entrée :
ci : integer : idem
- sorties :
long : integer : indicateur du nombre d'articles
num : integer : numéro de l'application cour-
rante
nom : string : nom de l'application
finfich : boolean : idem
- besoins :
fI17.data
f17a : file of r17 : fenêtre du fichier 17
- paramètre interne :
f : boolean : idem
- fonction :
lire le fichier 17.

293) - nom : liref25
cette procédure est entièrement équivalente à
liref17.

294) - nom :
liref24
- structure :
procédure
- mode d'appel :
liref24 (nvar, vd)
- entrée :
nvar : integer : idem
- sortie :
vd : boolean : valeur de la donnée
- besoin :
fI24.data
- paramètre interne :
f24a : file of array of boolean : fenêtre du
fichier 24
- fonction :
lire le fichier des variables de décision.

- 295) - nom :
liretl
- structure :
procédure
- mode d'appel :
liretl (nvar, ci, valeurlu, finfich)
- entrées :
nvar : integer : idem
ci : integer : idem
- sorties :
valeurlu : string : idem
finfich : boolean : idem
- besoin :
ftl : file of string : fenêtre sur le fichier de
texte libre
- paramètre interne :
f : boolean : idem
- fonction :
lire les fichiers de texte libre.
- 296) - nom :
liref21
- structure :
procédure
- mode d'appel :
liref21 (ci, lo, val1, val2, des, finfich)
- entrée :
ci : integer : idem
- sorties :
lo : integer : longueur de la description
val1 : string : donnée lue
val2 : string : donnée lue
des : tab : description lue
finfich : boolean : idem
- besoins :
fI21.data
f21a : file of r21 : fenêtre du fichier 21
- paramètres internes :
f : boolean : idem
i : integer : indice de boucle
- fonction :
lire le fichier 21.
- 297) - nom :
liref22
cette procédure est tout à fait identique à liref21.

- 298) - nom :
liref23
cette procédure est identique au deux précédentes.
- 299) - nom :
lireloca
- structure :
procédure
- mode d'appel :
lireloca (numappl, valeurlu, finfich)
- entrée :
numappl : integer : numéro de l'application
- sorties :
valeurlu : string : idem
finfich : string : idem
- besoin :
fo3 : boolean : indicateur si le fichier est ouvert
- paramètre interne :
s : string : reprend le numéro d'application
- fonction :
lire les localisations de services.
- 300) - nom :
liredescr
cette procédure est identique à la procédure
lireloca.
- 301) - nom :
liref26
- structure :
procédure
- mode d'appel :
liref26 (nvar, carlu)
- entrée :
nvar : integer : idem
- sortie :
carlu : char : le caractère lu
- besoin :
fI26.data
- paramètre interne :
f26a : file of array of char : fenêtre du fichier 26
- fonction :
lire le fichier 26.

- 302)- nom :
verif_name
- structure :
procedure
- mode d'appel :
verif_name (nom_cli, ok)
- entrée :
nom_cli : string : nom de la disquette à vérifier
- sortie :
ok : boolean : indicateur si tout va bien.
- utilise :
appel systeme
- fonction :
vérifie si la disquette de nom "nom_cli" est
bien montée.
- 303)- nom :
dern_q
- structure :
procedure
- mode d'appel :
dern_q (choice, num_fourn, num_q)
- entrées :
choice : char : détermine le nom du fichier
des réponses
num_fourn : integer : numéro d'un fournisseur
- sortie :
num_q : integer : numéro de la première question
à considérer, les valeurs étant :
0 : on part de la première question
-1 : toutes les réponses ont été
données
num_q : on part de la "num_q"ème question
- besoin :
fich_rep
- utilise :
reset, read, readln, strtoint, close
- paramètres internes :
fichier : string : nom du fichier à considérer
S1 : string : contient le nombre de questions
S2 : string : contient le numéro de la der-
nière question à laquelle on
a répondu
nq : integer : numéro de la dernière question
à laquelle on a répondu
nbreq : integer : nombre de questions
i : integer : variable de boucle
j : integer : variable de travail
k : integer : indice à partir duquel le
numéro de la dernière question
est contenu
car : char : un caractère lu
arrep : packed array [1..50] of char : une ligne
du fichier texte


```
- fonction :
  donne dans "num_q" le numéro de la première ques-
  tion à considérer (0 si elles sont toutes à con-
  sidérer, 1 si aucune n'est à considérer) pour
  le fournisseur numéro "num_fourn", et pour les
  questions suivant "choice" :
  choice : - Q : questions
            - A : questions relatives aux applica-
              tions
            - Y : questions relatives au matériel
- algorithme :
  (* choix du nom du fichier *)
  selon choice faire
    'Q' : fichier := '= 5 : REPQ.TEXT';
    'A' : fichier := '= 5 : REPA.TEXT';
    'Y' : fichier := '= 5 : REPY.TEXT';
  finsel;
  (* ouverture du fichier et lecture de la première
  ligne *)
  arrep := première ligne du fichier texte;
  (* initialiser S1 et S2 à "blanc" suivant le
  nombre de chiffres des numéros se trouvant dans
  arrep *)
  j := 1;
  pour i := 1 jusque 4 faire
    si arrep [i] ≠ '§' alors j := j+1 finsi finpour;
  selon j faire
    2 : S1 := ' ';          (* 1 blanc *)
    3 : S1 := '  ';        (* 2 blancs *)
    4 : S1 := '   ';       (* 3 blancs *)
    5 : S1 := '    ';      (* 4 blancs *)
  finsel;
  j := 1; k := (num_fourn x 4) + 1;
  pour i := k jusque k + 3 faire
    si arrep [i] ≠ '§' alors j := j + 1 finsi;
  finpour;
  selon j faire
    2 : S2 := ' ';
    3 : S2 := '  ';
    4 : S2 := '   ';
    5 : S2 := '    ';
  finsel;
  (* lecture dans arrep du nombre de questions *)
  j := 1;
  pour i := 1 jusque 4 faire
    si arrep [i] ≠ '§'
      alors - S1 [j] := arrep [i];
              - j := j + 1
    finsi
  finpour;
```

```

(* lecture dans arrep du numéro de la dernière
question à laquelle on a répondu pour le
num_fournème fournisseur *)
j := 1; k := (num_fourn x 4) + 1;
pour i := k jusqu' k + 3 faire
    si arrep [i] ≠ '$'
    alors - S2 [j] := arrep [i];
        - j := j + 1
    finsi
finpour;
(* transformation des strings en entiers *)
nbreq := strtoint (S1);
nq := strtoint (S2);
(* tests afin de vérifier ce qui est défini
dans sortie *)
si nq = 0
alors num_q := 0
sinon si nq ≠ nbreq
    alors num_q := nq + 1
    sinon num_q := -1
    finsi
finsi;
fermer le fichier fich_rep;

```

- 304) - nom :
- crit_dec_fx
 - structure :
 - fonction : boolean
 - mode d'appel :
 - crit_dec_fx (drive, num_rec)
 - entrées :
 - drive : string : identificateur d'un volume (e.g. '4')
 - num_rec : integer : numéro d'enregistrement (correspond à un numéro de fournisseur)
 - besoin :
 - fich_crit
 - utilise :
 - concat, reset, seek, get, close
 - paramètre interne :
 - fichier : string : nom du fichier utilisé
 - fonction :
 - cette fonction est à vrai si un critère décisif n'est pas satisfait, dans le cas contraire elle est à faux; et ce pour le fournisseur de numéro "num_rec".

- 305) - nom :
maj_crit_dec
- structure :
procedure
- mode d'appel :
maj_crit_dec (drive, num_rec)
- entrées :
drive : string : identificateur d'un volume
num_rec : integer : numéro d'enregistrement
(correspond à un numéro de
fournisseur)
- besoin :
fich_crit
- modifie :
fich_crit
- paramètre interne :
fichier : string : nom du fichier utilisé
- fonction :
elle stocke le fait que pour le fournisseur numéro
"num_rec" un critère décisif n'est pas satisfait.
- 306) - nom :
liregen
- structure :
procedure
- mode d'appel :
liregen (ndisk, choice, num_rec, ci, valintlu,
long, tableaulu, finfich)
- entrées :
ndisk : string : identificateur d'unité
(e.g. '4')
choice : char : détermine le fichier à lire
num_rec : integer : numéro de ligne où se position-
ner
ci : integer : détermine l'ouverture ou la
fermeture du fichier que l'on
désire consulter
- sorties :
valintlu : integer : numéro de question ou de
critère
long : integer : nombre de lignes de la ques-
tion ou du critère; lorsque
ci = 1, "long" correspond au
nombre de questions ou critères
tableaulu : tab20 : texte de la question ou du
critère
finfich : boolean : est à vrai lorsqu'on est au
bout du fichier

- besoin :
 - f4a
- utilise :
 - artltx, concat, reset, readln, close
- paramètres internes :
 - fichier : string : nom du fichier à consulter
 - st : string : ligne du fichier texte
 - f : boolean : est à vrai lorsqu'il faut fermer le fichier
 - i : integer : variable de boucle
 - k : integer : nombre de lignes + 1 d'une question
 - l : integer : variable de boucle
- fonction :

cette procédure lit sur l'unité identifiée par "ndisk", un fichier des questions ou des critères identifiés par "choice", à partir de la question située à la "num_rec"ème ligne si "ci" = 4, suivant les valeurs conventionnelles de "ci" (ci = 1 ouvrir le fichier, ci = 3 fermer le fichier); les informations lues sont le numéro de question (ou de critère) dans "valintlu", le nombre de lignes de la question (ou du critère) dans "long", le texte de la question (ou du critère) dans "tableaulu" et le fait que l'on est au bout du fichier dans "finfich". (Lorsque "ci" = 1, "long" signifie le nombre de questions ou de critères.)

où choice : 'S' = fichier des critères supplémentaires
'C' = fichier des critères personnalisés
'Q' = fichier des questions personnalisées
'A' = fichier des questions relatives aux applications
'Y' = fichier des questions synoptiques.

- 307) - nom :
 - artltx
- structure :
 - procedure
- mode d'appel :
 - artltx
- besoin :
 - f4a, long, valintlu, tableaulu, f
- modifie :
 - long, tableaulu, valintlu, f
- utilise :
 - readln, eof

- paramètres internes :
 - j : integer : variable de boucle
 - ll : integer : entier lu dans le fichier consulté
 - s : string : chaîne de caractères lue dans le fichier
- fonction :
 - lit effectivement pour la procédure liregen les valeurs "long", "tableaulu" et "valintlu"
- commentaire :
 - fait partie de liregen.

- 308)
- nom :
 - verif_crit
 - structure :
 - fonction : boolean
 - mode d'appel :
 - verif_crit (choice, num_rec)
 - entrées :
 - choice : char : détermine le nom du fichier
 - num_req : integer : numéro d'enregistrement, il correspond à un numéro de critère
 - besoin :
 - fich_stat
 - paramètres internes :
 - fichier : string : nom du fichier à considérer (fichier du statut des critères)
 - fonction :
 - la fonction est vraie si pour le "num_rec"ème critère ce critère est décisif; et ce pour les critères du type "choice"
 - où "choice" = - C : critères
 - S : critères supplémentaires.

- 309)
- nom :
 - sauv_stat
 - structure :
 - procedure
 - mode d'appel :
 - sauv_stat (ci, statut, choice)
 - entrées :
 - ci : integer : détermine l'ouverture ou la fermeture du fichier contenant le statut des critères
 - statut : boolean : statut d'un critère, "statut = true" si le critère est décisif
 - choice : char : détermine le nom du fichier des statuts des critères

- besoin :
fich_stat
 - modifie :
fich_stat
 - utilise :
rewrite, put, close
 - paramètre interne :
fichier : string : nom du fichier des statuts
des critères
 - fonction :
sauve le statut "statut" des critères (de manière
séquentielle) suivant les valeurs de ci :
ci = 1 : ouvrir le fichier et sauver;
2 : sauver le statut;
3 : fermer le fichier;
de plus choice détermine le type des critères :
choice = - C : critères
- S : critères supplémentaires.
- 310) - nom :
maj_crit_f
- structure :
procedure
 - mode d'appel :
maj_crit_f (num_f)
 - entrée :
num_f : integer : numéro d'enregistrement corres-
pondant à un numéro de fournis-
seur
 - besoin :
fich_crit
 - modifie :
fich_crit
 - utilise :
reset, seek, get, put, close
 - fonction :
stocke le fait que, pour le fournisseur numéro
"num_f", le codage de satisfaction des critères
est terminé.
- 311) - nom :
lire_crit_f
- structure :
procedure
 - mode d'appel :
lire_crit_f (num_f)
 - entrée :
num_f : integer : numéro d'enregistrement corres-
pondant à un numéro de four-
nisseur

- besoin :
fich_crit
 - utilise :
reset, seek, get, close
 - fonction :
la fonction est à vrai pour le fournisseur de
numéro "num_f" si le codage de satisfaction des
critères est terminé pour ce fournisseur.
- 312) - nom :
lire_conf_sup
- structure :
fonction : integer
 - mode d'appel :
lire_conf_sup (num_rec)
 - entrée :
num_rec : integer : numéro d'enregistrement corres-
pondant à un numéro de four-
nisseur
 - besoin :
fich_confs
 - utilise :
reset, seek, get, close
 - paramètres internes :
fichier : string : nom du fichier à considérer
(fichier contenant le nombre
de configurations par four-
nisseur)
 - fonction :
donne le nombre de configurations pour le four-
nisseur numéro "num_rec".
- 313) - nom :
maj_conf_sup
- structure :
procedure
 - mode d'appel :
maj_conf_sup (num_rec)
 - entrée :
num_rec : integer : numéro d'enregistrement cor-
respondant à un numéro de
fournisseur
 - besoin :
fich_confs
 - modifie :
fich_confs
 - utilise :
reset, seek, get, put, close

- paramètre interne :
 - fichier : string : nom du fichier à considérer (fichier contenant le nombre de configurations par fournisseur)
 - fonction :
 - incrémente de un le nombre de configurations pour le "num_rec"ème fournisseur.
- 314)
- nom :
 - sauv_com
 - structure :
 - procedure
 - mode d'appel :
 - sauv_com (l, coms, num)
 - entrées :
 - l : integer : nombre de lignes du commentaire
 - coms : tab3 : commentaires (minimum 7 lignes)
 - sortie :
 - num : car3 : numéro de l'enregistrement où la première ligne du commentaire est stockée
 - besoin :
 - fich_coms
 - modifie :
 - fich_coms
 - utilise :
 - reset, ioreult, strtoint, seek, get, put, str, length, close, rewrite
 - paramètres internes :
 - fichier : string : nom du fichier des commentaires
 - snum : string : chaîne de caractères de travail
 - i : integer : variable de travail
 - j : integer : variable de travail
 - inum : integer : variable de travail
 - fonction :
 - sauve le commentaire "coms" dont le nombre de lignes est "l", et donne le numéro de l'enregistrement où la première ligne est stockée dans "num".
- 315)
- nom :
 - sauv_rep
 - structure :
 - procedure
 - mode d'appel :
 - sauv_rep (num, num_fourn, ci, choice, repo, coms, l)

- entrées :
 - num : integer : numéro de ligne où se positionner
 - num_fourn : integer : numéro de fournisseur
 - ci : integer : détermine l'ouverture ou la fermeture du fichier des réponses
 - choice : char : détermine le nom du fichier des réponses
 - repo : string : code à stocker
 - coms : tab3 : commentaires relatifs à "repo"
 - l : integer : nombre de lignes du commentaire (l = 0 pour pas de commentaire)
- besoins :
 - fich_rep, sel_enreg
- modifie :
 - fich_rep
- utilise :
 - car3, reset, lire_conf_sup, read, strtoint, readln, write, sauv_com
- paramètres internes :
 - fichier : string : nom du fichier des réponses
 - s : string : chaîne de caractères lues
 - i, k, o : integer : variables de boucle
 - j : integer : nombre de configurations supplémentaires
 - s1 : packed array [1..6] of char : bloc de caractères lus
 - num_rec : packed array [1..4] of char : numéro d'enregistrement des commentaires
 - s2 : packed array [1..4] of char : bloc de caractères lus
 - car : char : caractères lus
- fonction :
 - cette procédure sauve à partir de la ligne "num", pour le fournisseur de numéro "num_fourn", selon "ci" (conventionnellement ci = 1, 2, 3), dans le fichier déterminé par "choice", le codage "repo" avec un commentaire "coms" de longueur "l" (si l = 0 il n'existe pas de commentaire)
 - où choice = 'C' : fichier du codage de satisfaction des critères personnalisés
 - 'S' : fichier du codage de satisfaction des critères supplémentaires
 - 'Q' : fichier du codage des réponses aux questions personnalisées
 - 'A' : fichier du codage des réponses aux questions informationnelles
 - 'Y' : fichier du codage des réponses aux questions synoptiques.

- 316) - nom :
cas3
- structure :
procédure
- mode d'appel :
cas3
- besoins :
fich_rep, choice, fichier, sel_enreg, o, car,
s2, s, k, i, j, num, num_fourn
- modifie :
fich_rep, o, car, s, s2, i, j, k
- utilise :
close, reset, lire_conf_sup, read, strtoint,
readln, str, length, write
- paramètre interne :
snum : string : chaîne de caractères intermédiaire
- fonction :
forme le cas "ci = 3" de la procédure sauv_rep.
- commentaire :
fait partie de sauv_rep.
- 317) - nom :
lire_compt
- structure :
procédure
- mode d'appel :
lire_compt (num_rec)
- entrée :
num_rec : integer : numéro d'enregistrement corres-
pondant à un numéro de four-
nisseur
- besoins :
fich_ev, com_ev
- modifie :
com_ev
- utilise :
reset, seek, get, close
- fonction :
stocke dans "com_ev" les compteurs pour le fournis-
seur de numéro "num_rec".
- 318) - nom :
sauver_compt
- structure :
procédure
- mode d'appel :
sauver_compt (num_rec)

- entrée :
 - num_rec : integer : numéro d'enregistrement correspondant à un numéro de fournisseur
 - besoins :
 - fich_ev, com_ev
 - modifiée :
 - fich_ev
 - utilise :
 - reset, seek, put, close
 - fonction :
 - stocke les compteurs du fournisseur de numéro "num_rec" sur fichier.
- 319)
- nom :
 - initcrit
 - structure :
 - procedure
 - mode d'appel :
 - initcrit (drive, ci)
 - entrées :
 - drive : string : identificateur de volume (e.g. '= 4')
 - ci : integer : détermine l'ouverture ou la fermeture du fichier considéré
 - besoin :
 - fich_crit
 - modifiée :
 - fich_crit
 - utilise :
 - concat, put, rewrite, close
 - paramètre interne :
 - fichier : string : nom du fichier à considérer (fichier de gestion des critères)
 - fonction :
 - initialise le fichier de gestion des critères sur le volume identifié par "drive"; cette initialisation est séquentielle et déterminée par ci (ci = 1, 2, 3 suivant les conventions).
- 320)
- nom :
 - init_confs
 - structure :
 - procedure
 - mode d'appel :
 - init_confs (drive)
 - entrée :
 - drive : string : identificateur de volume
 - besoins :
 - fich_confs, nfour2

- modifie :
fich_confs
- utilise :
concat, rewrite, put, close
- paramètres internes :
fichier : string : nom du fichier à considérer
(fichier contenant les compteurs)
- i : integer : variable de boucle
- fonction :
initialise le fichier contenant les compteurs de
configuration sur le volume identifié par "drive".

321) - nom :
initcomp

- structure :
procedure
- mode d'appel :
initcop (drive, ci)
- entrées :
drive : string : identificateur de volume
ci : integer : détermine l'ouverture ou la fer-
meture du fichier considéré
- besoins :
fich_ev, nfour2
- modifie :
fich_ev
- utilise :
concat, rewrite, put, seek, close
- paramètres internes :
enr_ev : ev : enregistrement d'initialisation
fichier : string : nom du fichier à considérer
(fichier contenant les compteurs)
- fonction :
initialise le fichier des compteurs sur le volume
identifié par "drive"; cette initialisation est
séquentielle et déterminée par ci (suivant les
conventions).

322) - nom :
init_rep

- structure :
procedure
- mode d'appel :
init_rep (choice, nbre_rec)
- entrées :
choice : char : détermine le nom du fichier
à considérer
nbre_rec : integer : nombre d'enregistrements prévus

- besoin :
fich_rep
- modifie :
fich_rep
- utilise :
str, rewrite, writeln, close
- paramètres internes :
fichier : string : nom du fichier des réponses
S : string : transformation de nbre_rec
reprint : packed array [1..50] of char : une
 ligne du fichier
reprint2 : packed array [1..50] of char : la
 première ligne du fichier
i, j, k : integer : variables de travail
- fonction :
initialise le fichier des réponses (réserve
"num_rec" enregistrements), le nom du fichier
étant déterminé par "choice"
où choice = - C : critères
 - S : critères supplémentaires
 - Q : questions
 - A : questions relatives aux applica-
 tions
 - Y : questions synoptiques

323) - nom :
verif

- structure :
fonction
- mode d'appel :
verif (choice, drive)
- entrées :
choice : char : détermine le fichier dont on
 vérifie l'existence
drive : string : détermine l'unité sur laquelle
 le fichier est
- utilise :
concat, reset, ioreult, close, erreur
- paramètres internes :
fich : text : fichier texte
f4a1 : file of r4 : fichier du type fi4
fichier1 : string : nom de fichier
fichier2 : string : nom de fichier
- fonction :
vérifie l'existence sur l'unité identifiée par
"drive" du fichier déterminé par choice
où choice = B : vérification de l'existence de B.D.A.
 P : vérification de l'existence du fi-
 chier des critères supplémentaires
 Q : vérification de l'existence du fi-
 chier des questions
 C : vérification de l'existence du fi-
 chier des critères
 S : vérification que la disquette sys-
 tème est bien présente.

- 324) - nom :
transf
- structure :
procedure
- mode d'appel :
transf (choice)
- entrée :
choice : char : détermine le fichier à transférer
- utilise :
reset, rewrite, seek, get, writeln, close, seek,
get, d_rem_files, eof
- paramètres internes :
- | | | |
|------------|------------------|--|
| f4a1 | : file of r4 | : fichier des questions relatives aux applications |
| ft1 | : file of string | : fichier des critères supplémentaires |
| fichier1 | : string | : nom du fichier à transférer |
| fichier2 | : string | : nom du fichier résultat |
| S | : string | : variable de travail |
| i | : integer | : variable de boucle |
| compt | : integer | : compteur d'enregistrements |
| f4a2, f4a3 | : text | : fichier de travail |
| result | : d_result | : résultat de la suppression de fichier dans le disque virtuel |
- fonction :
organise le transfert de fichier suivant choice :
- A : transfert du fichier des questions relatives aux applications (à partir volume 4)
 - Q : transfert du fichier des questions (à partir volume 9)
 - C : transfert du fichier des critères (à partir volume 9)
 - S : transfert du fichier des critères supplémentaires (à partir volume 4)

ANNEXE 12.16. : MODULE "TECHNIQUE"

- 325) - nom :
 strtoint
- structure :
 fonction : integer
- mode d'appel :
 strtoint (s)
- entrée :
 s : string : string à transformer en entier
- utilise :
 ord, length, exit, delete
- paramètres internes :
 z : integer : variable de travail
 k : integer : valeur absolue de l'entier
 i : integer : variable de travail d'une boucle
 signe : boolean : = vrai si entier négatif
- fonction :
 transforme le string "s" en un entier
- commentaire :
 l'algorithme de base est donné par Haut [7].
- 326) - nom :
 time
- structure :
 procedure
- mode d'appel :
 time (x)
- entrée :
 x : integer : nombre de secondes d'attente
- utilise :
 settimes, gettimes
- paramètres internes :
 heur, minute, second : integer
- fonction :
 effectue une attente de x secondes.
- 326b) - nom :
 bip
- structure :
 procedure
- mode d'appel :
 bip
- utilise :
 write, chr
- fonction :
 provoque l'émission d'une signal sonore.

- 327) - nom :
 encode
- structure :
 procedure
- mode d'appel :
 encode (password)
- entrée :
 password : string : mot de passe à encoder
- besoin :
 fich_pass
- modifie :
 fich_pass
- utilise :
 rewrite, length, ord, copy, concat, chr, writeln,
 close
- paramètres internes :
 Val : integer : valeur d'un code ascii
 N : integer : variable de travail d'une boucle
 code : string : mot de passe codé
 S : string : variable de travail
 car : string [1] : variable de travail
- fonction :
 encode le string "password" sur le fichier contenant le mot de passe (ce string doit être en lettres capitales).
- 328) - nom :
 decode
- structure :
 procedure
- mode d'appel :
 decode (password)
- sortie :
 password : string : mot de passe décodé
- besoin :
 fich_pass
- utilise :
 reset, length, ord, copy, concat, chr, readln,
 close
- paramètres internes :
 Val : integer : valeur d'un code ascii
 N : integer : variable de travail d'une boucle
 code : string : mot de passe décodé
 pass : string : mot de passe codé
 S : string : variable de travail
 car : string [1] : variable de travail
- fonction :
 donne le mot de passe décodé "password".

329) - nom :
transf_j_bord
- structure :
procédure
- mode d'appel :
transf_j_bord
- besoins :
fich_journ, fich2_journ
- modifie :
fich2_journ
- utilise :
reset, ioresult, rewrite, eof, seek, get, put,
close, bip, erreur, exit
- paramètre interne :
x : integer : numéro de record
- fonction :
transfère le journal de bord sur la disquette '= 5'.

330) - nom :
maj_journal
- structure :
procédure
- mode d'appel :
maj_journal (i)
- entrée :
i : integer : numéro de la procédure concernée
- besoin :
fich_journ
- modifie :
fich_journ
- utilise :
reset, seek, close, put, get, ioresult, erreur
- fonction :
met le switch de la procédure numéro "i" à 1.

330b) - nom :
consult_journal
- structure :
fonction : integer
- mode d'appel :
consult_journal (i)
- entrée :
i : integer : numéro de la procédure concernée
- besoin :
fich_journ
- utilise :
reset, seek, close, get, ioresult, erreur
- fonction :
détermine s'il faut exécuter la procédure numéro
"i", dans ce cas consult_journal = 0 (sinon = 1).

331) - nom :
dec_int
- structure :
procedure
- mode d'appel :
dec_int (Val. FVAL, SVAL)
- entrée :
Val : integer : entier à "décoder"
- sorties :
FVAL, SVAL : integer : premier et deuxième entier
- utilise :
mod, div
- paramètre interne :
iech : integer : valeur du shift à effectuer
- fonction :
donne, à partir de l'entier "Val", les deux entiers
"FVAL" et "SVAL" contenus dans "Val".

332) - nom :
enc_int
- structure :
procedure
- mode d'appel :
enc_int (FVAL, SVAL, VAL)
- entrées :
FVAL, SVAL : integer : entiers à "encoder"
- sortie :
VAL : integer : valeur du shift effectué
- paramètre interne :
iech : integer : valeur du shift effectué
- fonction :
stocke dans l'entier "Val" les entiers "FVAL"
et "SVAL".

334) - nom :
tab4_dec
- structure :
procedure
- mode d'appel :
tab4_dec (tabin, num)
- entrée :
tabin : car4 : tableau contenant un entier
(e.g. \$\$\$10)
- sortie :
num : integer : entier décodé
- utilise :
strtoint

- paramètres internes :
 - i, j : integer : variables de boucle
 - S : string : variable de travail
- fonction :
 - décode l'entier "num" codé dans le tableau "tabin".

335) - nom :
tabq_enc

- structure :
 - procedure
- mode d'appel :
 - tabq_enc (num, tabout)
- entrée :
 - num : integer : entier à insérer dans le tableau
- sortie :
 - tabout : car5 : tableau résultat du codage
- utilise :
 - str
- paramètres internes :
 - i : integer : variable de boucle
 - snum : string : variable de travail
- fonction :
 - positionne l'entier "num" dans le tableau "tabout" en laissant deux places vides à droite.

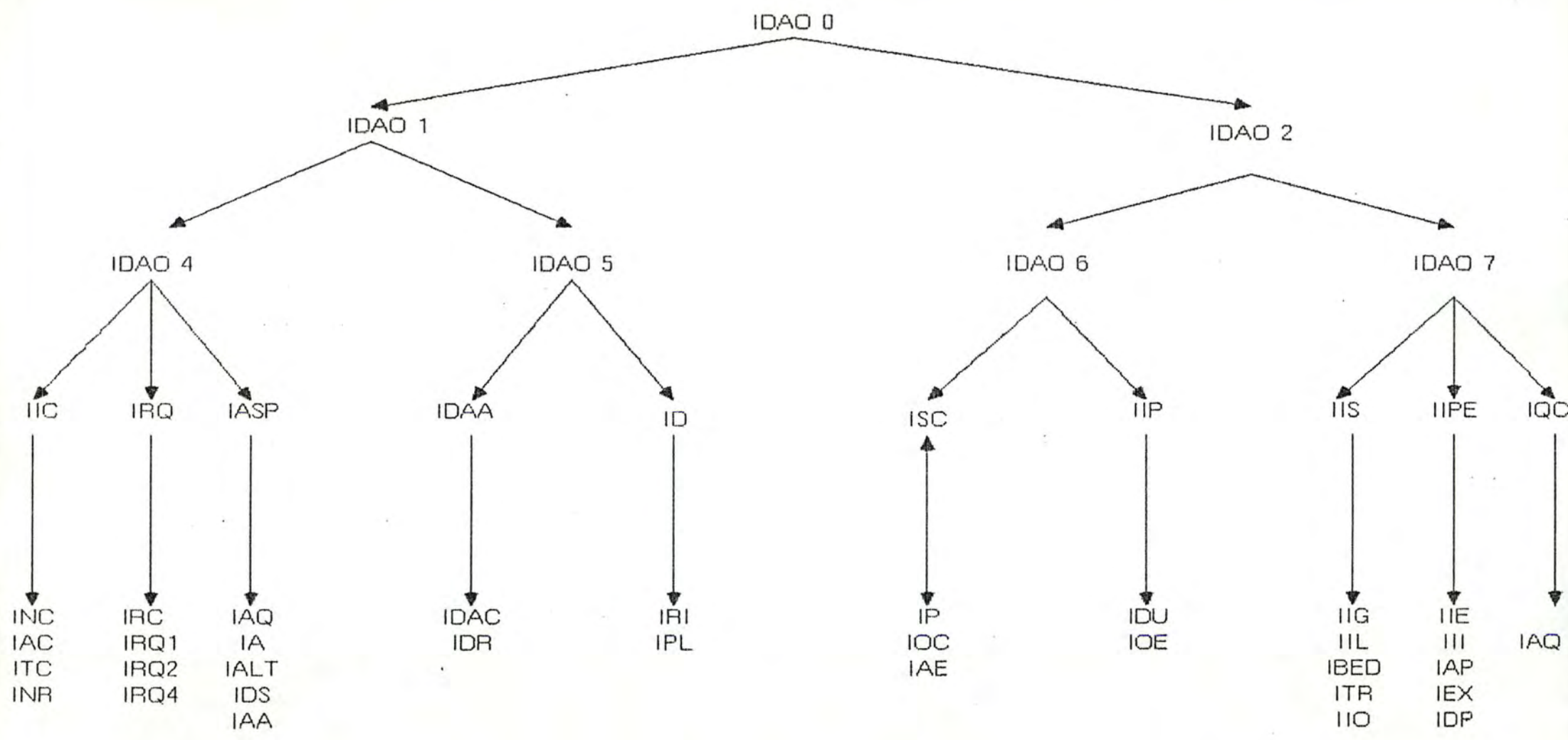
336) - nom :
tab_dec

- structure :
 - procedure
- mode d'appel :
 - tab_dec (tabin, code, l, num)
- entrée :
 - tabin : car6 : tableau dont il faut retirer les informations
- sorties :
 - code : char : code d'une réponse (i.e. 0, o, N, n, *, -)
 - l : integer : longueur d'un commentaire s'il existe
 - num : integer : numéro d'enregistrement du commentaire s'il existe
- utilise :
 - strtoint, tab4_dec
- paramètres internes :
 - S : string : variable de travail
 - tabtr : car4 : tableau de travail
 - i : integer : variable de boucle

- fonction :
décode un bloc de six caractères "tabin", les informations décodées sont le code de la réponse "code", le nombre de lignes "l" du commentaire s'il existe (sinon l = 0) et le numéro d'enregistrement "num" où il se trouve.
- 337) - nom :
tabc_enc
- structure :
procédure
 - mode d'appel :
tabc_enc (code, l, tabout)
 - entrées :
code : char : code d'une réponse
l : integer : si l = 0 alors pas de commentaire
 - sortie :
tabout : car4 : tableau résultat du codage
 - paramètre interne :
i : integer : variable de boucle
 - fonction :
positionne dans le tableau "tabout" le code d'une réponse déterminée par "code" qui est soit 0, o, N, n soit *, - et par "l" qui vaut 0 s'il n'existe pas de commentaires.
- 338) - nom :
tabf_enc
- structure :
procédure
 - mode d'appel :
tabf_enc (num, tabout)
 - entrée :
num : integer : entier à insérer dans le tableau
 - sortie :
tabout : car4 : tableau résultat du codage
 - paramètres internes :
i : integer : variable de boucle
snum : string : variable de travail
 - fonction :
positionne dans le tableau "tabout" l'entier "num" en laissant deux places vides à droite.

ANNEXE 13 : ARBRE DES PROCEDURES POUR L'INTRODUCTION
DES DONNEES DE L'APPEL D'OFFRES

Nous présentons dans cette annexe l'arbre des procédures relatives au module "Introduction des données de l'A.O."; ces procédures interviennent dans le journal de bord.



ANNEXE 14 : DICTIONNAIRE DES MESSAGES D'ERREUR

- e1 Fichier inconnu !! dans "envoie_message"
- e2 Fichier inconnu !! dans "recherche"
- e3 Erreur dans l'intervalle spatial !! dans "test_goto"
- e4 Fichier inconnu !! dans "trans"
- e5 Fichier inconnu !! dans "crea1"
- e6 Ancien password incorrect !! dans "lec_code"
- e7 Nouveau password incorrect !! dans "lec_code"
- e8 Incorrect password !! dans "acces_techn"
- e9 Ajout non permis !! dans "lire_a_coord"
- e10 Fichier inconnu !! dans "lister_enr"
- e11 Entier non valide !!
- e12 Société déjà existante !! dans "ISF"
- e13 Modification non permise !! dans "lire_m_coord"
- e14 Critère incorrect !! dans "list_fourn"
- e15 Société inconnue !! dans "M1E"
- e16 Fichier inconnu !! dans "supprimer_journal"
- e17 Fichier inconnu !! dans "lister_journal"
- e18 Pas de fournisseur !! dans "tr_fourn"; = pas de
fournisseur actif
- e19 Fichier inconnu !! dans maj_journal
- e20 Fichier inconnu !! dans "consult_journal"
- e21 Fournisseur inconnu !! dans "supp_fourn"
- e22 Pas de fournisseur !! dans "tr_fourn"
- e23 Fournisseur pas transféré !! dans "modif_av"
- e30 Fichier inconnu !! dans transf_j_bord
- e31 Heure invalide !! dans "IRI"
- e31bis Minute invalide !! dans "IRI"
- e32 Date invalide !! dans "liredate" --> jj
- e33 Date invalide !! dans "liredate" --> mm
- e34 Date invalide !! dans "liredate" --> ao

e36 Offre rejetée !! dans "dep_quest" et "pres_mat"
e37 Fournisseur inconnu !! dans "select"
e38 Aucune offre reçue !! dans "tr_fourn"
e39 Pas de fournisseurs !! dans "tr_fourn"
e40 Questions absentes !! dans "tr_quest"
e41 Pas la bonne disquette !! dans "tr_quest" et "tr_crit"
e42 Critères absents !! dans "tr_crit"
e43 Trop de configurations !! dans "pres_mat"
e50 Fichier résultat inexistant !! dans "Toolkit"
e51 Trop de fournisseurs !! dans "Tr_four"
e52 Trop de commentaire !! dans "Sauv_com"
e53 Transfert déjà effectué !! dans "Evaluate"

ANNEXE 15 : TABLES D'INFORMATIONS

Ces tables sont produites par un utilitaire qui s'appelle "XREF". La description complète de cet utilitaire est disponible dans le manuel fourni par Olivetti 12 page 6.13. Cependant, nous donnons ci-dessous une information sommaire sur les tables.

Dans la suite du texte, le terme "procédure" signifie procédure, fonction, programme,...

Pour des raisons de place nous donnerons que les différentes tables pour un module, à savoir le module technique.

ANNEXE 15.1. : TABLE 1

Cette table donne la liste de toutes les "procédures"
(avec leurs arguments) ainsi que le numéro de ligne
(absolu et relatif) où elles apparaissent.

Procedural Cross-Referencer - Version 5-02.IV.0.b

Absolute/Relative Program/Procedure/Function heading

2/ 2 unit technic;

24/ 24 function button (select : twobits) : boolean;

25/ 25 procedure paddle (select : twobits; var result : integer);

26/ 26 procedure note (pitch : pitch_range; octave : octave_range; duration : integer);

27/ 27 function lightpen (var charxpos, charypos, pixelxpos, pixelypos : integer) :
28/ 28 boolean;

29/ 29 procedure setkeys (tableptr : key_ptr);

30/ 30 procedure videomode (mode : integer);

31/ 31 procedure setfont (table : font_ptr);

32/ 32 procedure bkgnd_color (color : fourbits);

33/ 33 procedure palette (color : onebit);

34/ 34 procedure settime (hour, minute : integer);

35/ 35 procedure gettime (var hour, minute : integer);

36/ 36 procedure dummy1;

37/ 37 procedure dummy2;

38/ 38 procedure blink (onn : boolean);

39/ 39 procedure highlight (onn : boolean);

40/ 40 procedure foregnd (color : threebits);

41/ 41 procedure backgnd (color : threebits);

42/ 42 procedure video_page (pagenum : threebits);

43/ 43 procedure tocolor;

44/ 44 procedure tomono;

45/ 45 procedure set_attribute (ch : char);

46/ 46 procedure read_attribute (var ch : char);


```
47/ 47 procedure reverse_video;
48/ 48 procedure settimes (hour, minute, second : integer);
49/ 49 procedure gettimes (var hour, minute, second : integer);
50/ 50 function random : real;
51/ 51 procedure randomize;

103/ 103 procedure sc_use_info(do_what:sc_choice; var t_info:sc_info_type
;
104/ 104 procedure sc_use_port(do_what:sc_choice; var t_port:sc_tx_port);
105/ 105 procedure sc_erase_to_eol(x,line:integer);
106/ 106 procedure sc_left;
107/ 107 procedure sc_right;
108/ 108 procedure sc_up;
109/ 109 procedure sc_down;
110/ 110 procedure sc_getc_ch(var ch:char; return_on_match:sc_chset);
111/ 111 procedure sc_clr_screen;
112/ 112 procedure sc_clr_line (y:integer);
113/ 113 procedure sc_home;
114/ 114 procedure sc_eras_eos (x,line:integer);
115/ 115 procedure sc_goto_xy(x, line:integer);
116/ 116 procedure sc_clr_cur_line;
117/ 117 function sc_find_x:integer;
118/ 118 function sc_find_y:integer;
119/ 119 function sc_scrn_has(what:sc_scrn_command):boolean;
120/ 120 function sc_has_key(what:sc_key_command):boolean;
121/ 121 function sc_map_crt_command(var k_ch:char):sc_key_command;
122/ 122 function sc_prompt(line :sc_long_string; x_cursor,y_cursor,x_pos
where:integer; return_on_match:sc_chset;
123/ 123 no_char_back:boolean; break_char:char):char
124/ 124
125/ 125 function sc_check_char(var buf:sc_window; var buf_index,bytes_l
ft:integer)
126/ 126 :boolean;
```

```
127/ 127    function  sc_space_wait(flush:boolean):boolean;
128/ 128    procedure sc_init;
151/ 151    procedure init_curseur(num_mat:integer);
152/ 152    procedure curseur(num_mes,num_intro,num_mat:integer);
153/ 153    procedure efecran;
154/ 154    procedure ar_plan(f:threebits;x1,y1:integer);
155/ 155    procedure pr_plan(f:threebits;x1,y1:integer);
156/ 156    procedure calign_in(x1,y1:integer);
157/ 157    procedure calign_out(x1,y1:integer);
158/ 158    procedure message(num_mes,num_mat:integer;fich_mes:string);
159/ 159    procedure erreur(s:string);
160/ 160    procedure texte_libre(okset:setofchar;endroit:integer;var s:string
161/ 161        var d_endroit:integer);
162/ 162    procedure lirechar(okset:setofchar;var carlu:char);
163/ 163    function lirecar(okset:setofchar):char;
164/ 164    procedure lirestring(okset:setofchar;longueur:integer;var s:string
;
165/ 165    procedure lirereturn;
166/ 166    function lireentier:integer;
167/ 167    procedure lirecode(okset:setofchar;var s:string);
169/ 169    procedure openprinter;
170/ 170    procedure closeprinter;
171/ 171    procedure prpage;
172/ 172    procedure prsauteligne(i:integer);
173/ 173    procedure prfindepage(i:integer);
174/ 174    procedure prlongueur(i:integer);
175/ 175    procedure prcaracteres(i:integer);
176/ 176    procedure prlargeur(i:integer);
177/ 177    procedure prcondense;
178/ 178    procedure prelargi;
```



```
179/ 179 procedure praccentue;
180/ 180 procedure prdouble;
181/ 181 procedure prnormal;
183/ 183 procedure time(x:integer);
184/ 184 procedure bip;
185/ 185 function strtoint(s:string):integer;
186/ 186 procedure encode(password:string);
187/ 187 procedure decode(var password:string);
188/ 188 procedure transf_j_bord;
189/ 189 procedure maj_journal(i:integer);
190/ 190 function consult_journal(i:integer):integer;
191/ 191 procedure dec_int(val:integer;var fval,sval:integer);
192/ 192 procedure enc_int(fval,sval:integer;var val:integer);
208/ 208     procedure test_goto(x,y:integer);
225/ 225     procedure init_curseur;
308/ 308     procedure curseur;
328/ 328     procedure efecran;
340/ 340     procedure ar_plan;
356/ 356     procedure pr_plan;
372/ 372     procedure clign_in;
388/ 388     procedure clign_out;
404/ 404     procedure recherche(num_mes:integer;fich_mes:string;var rech1,
ech2:integer);
442/ 442     procedure envoie_message(rech1,rech2:integer;fich_mes:string);
478/ 478     procedure message;
501/ 501     procedure erreur;
520/ 520     procedure texte_libre;
557/ 557     procedure lirechar;
576/ 576     function lirecar;
```



```
614/ 614      procedure lirestring;
689/ 689      procedure lirereturn;
705/ 705      function lireentier;
774/ 774      function lirecar2(okset:setofchar):char;
801/ 801      procedure lirecode;
841/ 841      procedure encode;
891/ 891      procedure decode ;
943/ 943      procedure transf_j_bord;
978/ 978      procedure maj_journal;
1009/ 1009    function consult_journal;
1048/ 1048    procedure dec_int;
1064/ 1064    procedure enc_int;
1080/ 1080    procedure openprinter;
1093/ 1093    procedure closeprinter;
1106/ 1106    procedure prpage;
1118/ 1118    procedure prsauteligne;
1132/ 1132    procedure prfindepage;
1145/ 1145    procedure prlongueur;
1157/ 1157    procedure prcaracteres;
1170/ 1170    procedure prlargeur;
1183/ 1183    procedure prcondense;
1196/ 1196    procedure prelargi;
1209/ 1209    procedure praccentue;
1222/ 1222    procedure prdouble;
1234/ 1234    procedure prnormal;
1246/ 1246    procedure time;
1263/ 1263    procedure bip;
1273/ 1273    function strtoint;
1367/ 1367    end.
```

ANNEXE 15.2. : TABLE 2

Cette table donne la liste de toutes les "procédures" par ordre alphabétique, et pour chaque "procédure" elle donne :

- le numéro de ligne de la déclaration,
- le numéro de ligne où se trouve le "begin",
- le numéro de ligne où le bloc de la "procédure" commence (si procédure est déclarée "forward"),
- le nom de la "procédure",
- une liste des "procédures" appelées (dans l'ordre d'occurrence).

39			ext	highligh :					
151	235	225	initcur	: reset	ioresult	seek	get	close	
				: eof	put	rewrite			
27			ext	lightpen :					
163	586	576	lirecar	: read	eoln	chr	write		
774	781		lirecar2	: read	eoln	chr	write		
162	567	557	lirechar	: lirestri					
167	813	801	lirecode	: length	lirecar2	concat	delete		
166	718	705	lireenti	: ord	length	lirecar	trunc	abs	
				: write	concat	strtoint	delete		
165	698	689	lireretu	: lirechar					
164	628	614	lirestri	: scfind	write	gotoxy	length	lirecar	
				: concat	delete				
189	985	978	majjour	: reset	ioresult	seek	get	put	
				: close	erreur				
158	489	478	message	: initcur	decint	gotoxy	recherch	envoien	
26			ext	note :					
169	1086	1080	openprin	: close	reset				
25			ext	paddle :					
33			ext	palette :					
179	1215	1209	praccent	: write					
175	1163	1157	prcaract	: write	chr				
177	1189	1183	prconden	: write					
180	1228	1222	prdoubled	: write					
178	1202	1196	prelargi	: write					
173	1139	1132	prfindep	: write	chr				
176	1176	1170	prlargeu	: write	chr				
174	1151	1145	prlongue	: write	chr				
181	1240	1234	prnormal	: write					
171	1112	1106	prpage	: page					

155	362	356	prplan	:	testgot	gotoxy	foregnd		
172	1126	1118	prsautel	:	writeln				
50			ext random	:					
51			ext randomiz	:					
46			ext readatt	:					
404	415		recherch	:	concat	copy	length	reset	ioresul
				:	seek	get	close	erreur	exit
47			ext reverse	:					
125			ext sccheck	:					
116			ext scclrc	:					
112			ext scclrl	:					
111			ext scclrs	:					
109			ext scdown	:					
114			ext sceras	:					
105			ext scerase	:					
117			ext scfind	:					
110			ext scgetc	:					
115			ext scgoto	:					
120			ext schask	:					
113			ext schome	:					
128			ext scinit	:					
106			ext sclleft	:					
121			ext scmapc	:					
122			ext scprompt	:					
107			ext scright	:					
119			ext scscrn	:					
127			ext scspace	:					
108			ext scup	:					
103			ext scusei	:					

104		ext	scusep	:						
45		ext	setattr	:						
31		ext	setfont	:						
29		ext	setkeys	:						
34		ext	settime	:						
48		ext	settimes	:						
185	1282	1273	strtoint	:	ord	length	exit	strtoint	delete	
2	1313		technic	:	chr					
208	214		testgot	:	erreur	exit				
160	529	520	texteli	:	sceras	gotoxy	write	lirestri		
183	1254	1246	time	:	settimes	gettimes				
43		ext	tocolor	:						
44		ext	tomono	:						
188	952	943	transfj	:	reset	ioresult	rewrite	eof	seek	
				:	get	put	close	bip	erreur	
				:	exit					
30		ext	videomod	:						
42		ext	videopa	:						

ANNEXE 15.3. : TABLE 3

Cette table donne par ordre alphabétique la liste des "procédures", et pour chacune d'elles les "procédures" qui l'appellent.

Procedural Cross-Referencer - Version S-02.IV.0.b

Procedure	Called By				
abs	:lireenti				
backgnd	:arplan				
bip	:transfj				
blink	:clignin	clignou	erreur		
chr	:bip :lirecar2 :technic	decode prcaract	efecran prfindep	encode prlargeu	lirecar prlongue
close	:closepri :initcur	consult majjour	decode openprin	encode recherch	envoiem transfj
concat	:decode :recherch	encode	lirecode	lireenti	lirestri
copy	:decode	encode	recherch		
decint	:curseur	message			
delete	:lirecode	lireenti	lirestri	strtoint	
envoiem	:message				
eof	:initcur	transfj			
eoln	:lirecar	lirecar2			
erreur	:consult :transfj	envoiem	majjour	recherch	testgot
exit	:envoiem	recherch	strtoint	testgot	transfj
foregnd	:prplan				
get	:consult :transfj	envoiem	initcur	majjour	recherch
gettimes	:time				
gotoxy	:arplan :erreur	clignin lirestri	clignou message	curseur prplan	efecran texteli
initcur	:curseur	message			
ioresult	:consult :transfj	envoiem	initcur	majjour	recherch
length	:decode :recherch	encode strtoint	lirecode	lireenti	lirestri

lirecar	:lireenti	lirestri			
lirecar2	:lirecode				
lirechar	:lireretu				
lirestri	:lirechar	texteli			
ord	:decode	encode	lireenti	strtoint	
page	:prpage				
put	:initcur	majjour	transfj		
read	:lirecar	lirecar2			
readin	:decode				
recherch	:message				
reset	:closepri :majjour	consult openprin	decode recherch	envoiem transfj	initcur
rewrite	:encode	initcur	transfj		
sceras	:texteli				
scerase	:erreur				
scfind	:lirestri				
seek	:consult :transfj	envoiem	initcur	majjour	recherch
settimes	:time				
strtoint	:lireenti	strtoint			
testgot	:arplan	clignin	clignou	prplan	
time	:erreur				
trunc	:lireenti				
write	:bip :lireenti :prdouble :prnormal	efecran lirestri prelargi texteli	erreur praccnt prfindep	lirecar prcaract prlargeu	lirecar2 prconden prlongue
writeln	:encode	envoiem	prsautel		

ANNEXE 15.4. : TABLE 4

Cette table donne une liste alphabétique de "procé-
dures"; pour chaque "procédure" la table donne les
variables que la "procédure" examine ou modifie, si la
variable est modifiée elle est précédée d'une astéris-
que (*), si la variable n'est pas locale à la "procédure"
le nom de la "procédure" dans laquelle elle est déclarée
est donné.

Procedural Cross-Referencer - Version S-02.IV.0.b

Procedure	References	Variables
arplan :	f	x1
:	y1	
backgnd :		
bip :		
bkgndco :		
blink :		
button :		
clignin :	x1	y1
clignou :	x1	y1
closepri :		
consult :	technic's fichjou	i
:	technic's *routage	
curseur :	technic's matcur	numintr
:	nummat	nummes
:	x	x1
:	y	y1
decint :	*fval	*iech
:	*sval	val
decode :	*car	*code
:	technic's fichpas	*n
:	pass	*password
:	*s	*val
dummy1 :		
dummy2 :		
efecran :		
encint :	fval	*iech
:	sval	*val
encode :	*car	*code
:	technic's fichpas	*n
:	password	*s
:	*val	
envoie :	technic's fichrecr	fichmes
:	*rech1	rech2


```

erreur      :          $

foregnd     :

gettime     :

gettimes    :

highlight   :

initcur     :          *ficheur          *i
            :          *j              technic's *matcur
            :          nummat            technic's *nummatc

lightpen    :

lirecar     : technic's bell          technic's bs
            :          *ch              technic's ensacc
            :          *ok                  okset

lirecar2    : technic's bell          *ch
            :          *ok                  okset

lirechar    :          *carlu          *longueur
            :          okset                stringlu

lirecode    : technic's bs          *car
            : technic's cr          *l
            :          *lireset         *long
            :          okset              *s
            :          *s1

lireenti    : technic's bell          technic's bs
            :          *car              technic's cr
            :          *lireset         *longueur
            :          *maxint          technic's numset
            :          *okset           *s
            :          technic's signeset *valeur
            :          *z

lireretu    :          ch          technic's cr
            :          *okset

lirestri    : technic's bs          *car
            : technic's cr          *i
            :          *l            *lireset
            :          *long         longueur
            :          okset        *s
            :          *s1           *x
            :          *y

majjour     : technic's fichjou      i

message     :          fichmes        technic's matcur
            :          nummat                nummes
            :          rech1                  rech2
            :          x                      y

```

note :
openprin :
paddle :
palette :
praccent : technic's em
prcaract : technic's cs i
prconden : technic's co
prdouble : technic's d0
prelargi : technic's el
prfindep : i technic's so
prlargeu : technic's cw i
prlongue : technic's fl i
prnormal : technic's no
prpage :
prplan : f x1
: y1
prsautel : i *k
random :
randomiz :
readatt :
recherch : technic's annexa *fichanx
: fichmes nummes
: *rech1 *rech2
reverse :
sccheck :
scclrc :
scclrl :
scclrs :
scdown :
sceras :

scerase :
sfind :
sgetc :
sgoto :
shask :
shome :
scinit :
sleft :
smapc :
spromp :
sright :
sscrn :
sspace :
scup :
scusei :
scusep :
setattr :
setfont :
setkeys :
settime :
settimes :
strtoint : *i *k
: s *signe
: *z

technic : *alphaset *bell
: *bs *cr
: *ensacc *enscar
: *enschiff *ensemble
: *enslettr *nummatc
: *numset *routage
: *signeset

testgot : x y

ANNEXE 15.5. : TABLE 5

Cette table donne une liste alphabétique des variables et pour chaque variable elle donne :

- le nom de la "procédure" où elle est déclarée;
- la liste des "procédures" qui l'examinent ou la modifient.

Procedural Cross-Referencer - Version S-02.IV.0.b

Procedure	Variable	Referenced By
technic	alphaset :technic	
technic	anxecra :recherch	
technic	bell :lirecar	lirecar2 lireenti technic
technic	bs :lirecar	lirecode lireenti lirestri technic
decode	car :decode	
encode	car :encode	
lirecode	car :lirecode	
lireenti	car :lireenti	
lirestri	car :lirestri	
lirechar	carlu :lirechar	
lirecar	ch :lirecar	
lirecar2	ch :lirecar2	
lireretu	ch :lireretu	
technic	co :prconden	
decode	code :decode	
encode	code :encode	
technic	cr :lirecode	lireenti lireretu lirestri technic
technic	cs :prcaract	
technic	cw :prlargeu	
technic	d0 :prdouble	
texteli	dendroi :texteli	
technic	el :prelargi	
technic	em :praccent	
texteli	endroit :texteli	
technic	ensacc :lirecar	technic
technic	enscar :technic	

technic enschiff :technic
technic ensemble :technic
technic enslettr :technic
arplan f :arplan
prplan f :prplan
technic fich2jo :transfj
recherch fichanx :recherch
technic fichjou :transfj
recherch fichmes :recherch
time hour :time
prsautel i :prsautel
strtoint i :strtoint
prsautel k :prsautel
strtoint k :strtoint
time minute :time
transfj n :transfj
technic nummatc :technic
recherch nummes :recherch
technic numset :technic
texteli okset :texteli
recherch rech1 :recherch
recherch rech2 :recherch
technic routage :technic
strtoint s :strtoint
texteli s :texteli
time second :time
strtoint signe :strtoint
technic signeset :technic

testgot	x	:testgot
time	x	:time
prplan	x1	:prplan
testgot	y	:testgot
prplan	y1	:prplan
strtoint	z	:strtoint

ANNEXE 16 : GUIDE D'UTILISATION DE L'INFORMATICIEN

Nous décrivons dans cette annexe la configuration informatique choisie, puis nous expliquons comment rendre un programme exécutable.

Nous donnons également la liste des "units" et utilitaires du système que nous avons employés.

Pour terminer nous documentons le catalogue des disquettes utilisées pour le logiciel.

ANNEXE 16.1. : CONFIGURATION INFORMATIQUE

Nous avons programmé en PASCAL-UCSD sur OLIVETTI M24. Ce langage a été choisi pour des raisons de portabilité et de facilité.

Nous avons rencontré quelques problèmes. En effet, l'éditeur de liens ne fonctionne pas et lors de l'exécution d'un "close (filename, lock)" nous obtenons un article supplémentaire dans le fichier "filename".

Afin de contourner ces deux problèmes, nous avons d'une part utilisé la librairie ("system.library") et d'autre part créé une gestion de pointeurs pour détecter la fin des fichiers utilisés (cependant, dans certains cas l'algorithme d'accès aux fichiers a simplement été adapté en tenant compte de cet article supplémentaire).

Un autre problème est apparu (ce n'est pas un problème de langage) : en effet, il n'est pas possible, lors de l'exécution d'un programme, de déterminer si l'imprimante est sous tension ou pas. Il est donc important de s'assurer que l'imprimante est mise sous tension avant d'activer une fonction requérant une impression.

Nous ne regrettons pas d'avoir choisi le langage PASCAL-UCSD malgré ses problèmes. En effet, que pouvions-nous choisir comme autre langage sur Olivetti M24 ?

Certains langages ne convenaient pas au problème à résoudre (gestion d'écrans, gestion de fichiers,...) comme par exemple : FORTRAN, LISP, PROLOG,...

Il restait donc les langages C et turbo-PASCAL.

Nous avons effectué pour le cours de "Performances et mesures" de Monsieur Ramaekers un travail : nous avons étudié la qualité des messages d'erreurs et de la documentation de plusieurs compilateurs de C dont celui de l'Olivetti M24. Notre conclusion pour l'Olivetti fut de loin la plus négative. De plus, cette implémentation du langage C proposait cinq compilateurs, d'où une question se posait : lequel choisir ? Nous avons donc laissé à côté ce langage.

Pour le langage turbo-PASCAL, nous avons estimé qu'il fallait le laisser de côté : il n'offre pas la structure de "unit", lors de chaque erreur la compilation est arrêtée,...

ANNEXE 16.2. DEMARCHE POUR RENDRE EXECUTABLE UN PROGRAMME

Supposons avoir à rendre exécutable un programme utilisant des "unit".

Si toutes les "unit" se trouvent déjà en librairie, alors il suffit simplement de compiler le programme, sinon il faut les y introduire avant de compiler le programme. Pour introduire des "units" en librairie, il faut d'abord les compiler puis exécuter le programme "library.code".

Exemple : supposons avoir A un programme et B, C deux "units", nous donnons ci-dessous les étapes à suivre si A utilise B et B utilise C :

- compiler C;
- introduire C en librairie;
- réinitialiser le système;
- compiler B;
- introduire B en librairie;
- réinitialiser le système;
- compiler A.

Notons que B est compilé APRES que C ait été introduit en librairie, il faut d'abord introduire en librairie les "units" de niveau le plus bas et puis remonter la hiérarchie.

Notons encore que pour rendre un programme auto-exécutable, il suffit de renommer le fichier "filename.code" en "system.startup" ou "system.menu".

ANNEXE 16.3. : UTILISATION DU SYSTEME

Nous avons utilisé des "unit" du système dont la documentation est détaillée dans un manuel fourni par OLIVETTI 12 . Ces "units" se nomment : screenop, IBM_SPECIAL, WILD, DIRINFO, COMMANDIO.FILE.INFO.

Remarquons que pour WILD et DIRINFO, les "unit" qui se trouvaient à l'origine en librairie n'étaient pas valables, il a fallu les introduire "à la main" i.e. comme une autre "unit".

Nous avons également utilisé des utilitaires qui sont expliqués dans un manuel fourni par OLIVETTI 12 : XREF, LIBRARY.

ANNEXE 16.4. : CONTENU DES DISQUETTES

Nous donnons tout d'abord pour chaque programme la liste des fichiers qu'il faut sur une disquette pour que l'on puisse exécuter le programme.

Ensuite pour chaque disquette nous donnons le nom des fichiers ainsi que deux significations.

Programme "fournisseur"

system.pascal; system.miscinfo; system.pme.86; system.config;
ecran_fo.data; a_ecran_fo.data; curseur.data; system.library;
system.startup où :

- le system.library comprend en plus de la librairie habituelle : screenop.code, technic.code
- le system.startup est le programme "fournisseur".

Programme "intro"

system.pascal; system.miscinfo; system.pme.86; system.config;
ecran_ao.data; a_ecran_ao.data; curseur.data; journal.data;
password.text; system.library; system.startup; system.menu;
questmod.text; redaction.code; redac2.code; textes.text;
lint.text; où :

- le system.library comprend en plus de la librairie habituelle : screenop.code, technic.code, toolkit.code, BD1.code, initial.code, impression.code
- le system.startup est le programme "start"
- le system.menu est le programme "intro".

Programme "evaluate"

system.pascal; system.miscinfo; system.pme.86; system.config;
ecran_ev.data; a_ecran_eo.data; curseur.data; system.library;
system.startup; fi40.text; questi.text; criti.text où :

- le system.library comprend en plus de la librairie habituelle : screenop.code, technic.code

- le system.startup est le programme "evaluate".

Disquette "FOURN:"

- system.pascal; system.miscinfo; system.pme.86;
system.config : systèmes fournis par Olivetti.
- system.library : contient, en plus de la librairie habituelle, screenop.code et technic.code.
- ecran_fo.data : fichier des écrans transformé en fichier d'enregistrements.
- a_ecran_fo.data : fichier de "pointeurs" relatifs à ecran_fo.data.
- curseur.data : fichier contenant les matrices de coordonnées des curseurs.
- system.startup : compilation de "fourn.text".

Disquette "LIAO:"

- system.pascal; system.miscinfo; system.pme.86;
system.config : systèmes fournis par Olivetti.
- system.library : contient, en plus de la librairie habituelle, screenop.code, technic.code, toolkit.code, BD1.code (compilation de "BD1a.text"), initial.code, impression.code.
- ecran_ao.data : fichier des écrans transformé en fichier d'enregistrements.
- a_ecran_ao.data : fichier de "pointeurs" relatifs à ecran_ao.data.
- curseur.data : fichier contenant les matrices de coordonnées des curseurs.
- journal.data : fichier contenant le journal de bord.
- password.text : fichier contenant le mot de passe.
- system.startup : compilation de "start.text".
- system.menu : compilation de "introao.text".
- redac2.code : compilation de "redac2.text"
- redaction.code : compilation de "redaction.text".
- textes.text : texte anonyme de l'appel d'offres.

- questmod : questionnaire.
- lint : texte de la lettre d'intention anonyme.

Disquette "EVAL:"

- system.pascal; system.miscinfo; system.pme.86;
system.config : systèmes fournis par Olivetti.
- system.library : contient, en plus de la librairie habituelle, screenop.code, technic.code.
- ecran_eo.data : fichier des écrans transformé en fichier d'enregistrements.
- a_ecran_eo.data : fichier de "pointeurs" relatifs à ecran_eo.data.
- curseur.data : fichier contenant les matrices de coordonnées des curseurs.
- system.startup : compilation de "evaluate.text".
- fi40.text : fichier des questions synoptiques.
- questi.text : fichier des questions anonymes.
- criti.text : fichier des critères anonymes.

Disquette "MEM1:"

- questmod.text : questionnaire de l'appel d'offres
- lint.text : lettre d'intention anonyme
- textes.text : appel d'offres anonymes
- questi.text : questionnaire de l'appel d'offres
- criti.text : critères de l'appel d'offres
- fi40.text : questions synoptiques
- ecran_ao.text : fichier des écrans relatifs à l'introduction des données
- ecran_fo.text : fichier des écrans relatifs à la gestion des fournisseurs
- ecran_eo.text : fichier des écrans relatifs à l'évaluation
- start.text : programme d'attente
- technic.text : "unit" technic

- toolkit.text : "unit" toolkit
- redaction.text : programme de rédaction de l'appel
d'offres
- redac2.text : programme de rédaction de la
lettre d'intention
- impression.text : unit redacprint
- fourn.text : programme fournisseur
- introao.text : programme intro (introduction
des données)
- intro2.text : suite de introao.text

Disquette "MEM2:"

- initial.text : "unit" init
- BD1a.text : "unit" BD1
- BD1b.text : suite de BD1a.text
- evaluate.text : programme evaluate
- evaluate2.text : suite de evaluate.text
- evaluate3.text : suite de evaluate2.text

acces_rechn	83	ecritf11	263
aj_arb_f	182	ecritf12	264
ajout_curseur	85	ecritf13	265
ajout_fourn	3	ecritf16	266
ajout_journal	92	ecritf17	267
an_crit	118	ecritf21	269
analy	155	ecritf22	270
annexe1	81	ecritf23	271
ar_pian	197	ecritf24	274
arb_f	184	ecritf25	268
art	237	ecritt1	253
art	239	efecran	195
artlu	281	efface	240b
artlu	284	enc_int	352
artlux	307	encode	327
augm_compt	115	envoi_message	203
bip	326b	erbis	252b
calcoster	72	erreur	204
cas3	316	etat	188
chng_disk	257	exit_fourn	12b
choice	242	exterieur	179
clign_in	199	f_deja_intro	130
clign_out	200	f_existe	131
closeprinter	215	fermer	175
codage	123	fichii	170
consult_journal	330b	fichresulao	141
crea_anl	80	fin_lig	143
crea_arb_f	180	help_coo	100
crit_dec_fx	304	help_editeur	99
curseur	194	help_ev	101
dec_int	331	help_i_texte	98
decalage	151	help_system	97
decode	328	i_act	76
dep_quest	117	i_applic	34
dera_g	303	ia	45
det_arb_f	181	iaa	55
dta_ens_cod	121	iac	38
dta_ens_menu	122	iae	67
ecrdescr	273	iaf	6
ecriecran	212	ialt	46
ecrf11b	277	iap	64
ecrf26	278	iaq	54
ecrfourn	244	iasp	25
ecrloca	272	ibed	59
ecrnum	248	id	26
ecrire	144	idaa	29
ecrit	275	idac	69
ecritf1	254	idao0	16
ecritf2	255	idao1	17
ecritf4	258	idao2	18
ecritf6	260	idao4	19
ecritf7	261		

idao5	20	insf7	165
idao6	21	insf11	166
idao7	22	insf12	167
idate	75	insf13	168
idc	53	insf16	169
ide	51	insf17	157
idp	65	insf21	160
idr	57	insf22	159
ids	47	insf23	161
idu	50	insf25	158
iox	60	inter	171
if1	227	intro_ao	14
if2	228	intro_li	73
if7	229	imp_ac	96d
if12	230	impacc	178
if13	231	imp_f	186
if16	232	ip	48
if17	233	ipl	49
if21	234	iga	68
if22	235	igc	32
if23	236	irc	41
if24	237	iri	66
if25	238	irq	24
if26	239	irq1	42
iic	23	irq2	43
iie	42	irq3	44
iif	4	isc	28
iig	52	isf	5
iii	63	itc	39
iil	56	itr	58
io	61	lec_code	82
iip	27	liberer	96
iipe	31	ligbia	152
iis	30	lire	145
inc	37	lire_a_coord	86
info	142	lire_compt	317
init	241	lire_conf_sup	312
init_confs	320	lire_crit_f	311
init_curseur	193	lire_m_cord	88
init_et_av	9	lirebfourn	250
init_rep	322	lirecar	208
initcrit	319	lirecar2	210b
initcomp	321	lirecnar	206
initetat	247	lirecodage	213
inr	40	lirecode	210c
insf1	162	lirecom	211
insf2	163	liredate	71b, 36
insf6	164	liredescr	300

lireentier	210	modifourn	245
liref1	279	modif_av	9
liref2	280	modif_cur	87
liref4	282	mte	11
liref6	285	nrappi	27e
liref7	283	num	153
liref11	287	num1	153
liref11b	288	openprinter	214
liref12	289	ouvert	172
liref13	290	parc_arb_f	189
liref16	291	parc_arb_num	190
liref17	292	patience	240
liref21	296	pgfoot	175
liref22	297	pghead	174
liref23	298	pr_plan	198
liref24	294	praccentue	224
liref25	293	prcaracteres	220
liref26	301	prcondense	222
lirefourn	249	prdouble	225
liregen	306	prelargi	223
lireloca	299	pres_bl	33
lirenum	250b	pres_mat	120
lirereturn	209	prfindepage	218
lirestring	207	priargeur	221
liret1	295	prlongueur	219
lirnumb	246	prnormal	226
list_ac	96b	prpage	216
list_arb_f	185	prsauteligne	217
lister_cur	90	quest	149
list_fourn	7	recherche	202
list_li	96c	reserver	95
list_sel_f	191	resulb	156
lister_f	187	sautpg	176
lister_journal	94	sauv_com	314
mle	10	sauv_rep	315
maj_com	116	sauv_stat	309
maj_conf_sup	313	sauver_compt	318
maj_crit_dec	305	sel_fourn	192
maj_crit_f	310	select	124
modif_cur	87	soulig	151
menu_curseur	84	stat_crit	119
menu_eval	129	strint	252
menu_fourn	1	strtoint	325
menu_journal	91	sup_arb_f	183
menu_toolkit	77	sup_cur	89
meaf	12	supp_fourn	13
message	201	supprimer_journal	93
maj_journal	330		

susp_ao	71
susp_ev	114
t_e_crit	106
t_e_quest	105
t_e_syn	107
tab4_dec	334
tab_choix	133
tab_crea	132
tab_crit	110
tab_dec	336
tab_e_c	138
tab_f_lig	135
tab_fe_lig	140c
tab_fin	112
tab_fo	108
tab_impr	102
tab_mat	109
tab_num	104
tab_of_rej	140b
tab_sup	103
tab_q_lig	136
tab_quest	111
tab_reorg	113b
tabc_enc	337
tab_comp	139
tabf	134
tabf_enc	338
tabl	146
tableaux	113
tabq_enc	335
tabt_enc	137
test_goto	196
text_data	79
texte_libre	205
time	326
tr_crit	127
tr_four	125
tr_fourn	2
tr_quest	126
trans	78
trans	324
transcom	140
transfert	128
trans_j_bord	329
trsf	177
txlb	148
vardec	154
verif	323
verif_crit	308
verif_fourn	243
verif_j_b	70, 35
verif_name	302
welc_ao	15
welc_li	74

ANNEXE 18 : PROGRAMMES SOURCES

Les programmes sources sont disponibles sur demande
auprès de Monsieur van Bastelaer.

PARTIE IX

BIBLIOGRAPHIE

- [1] J.P. Adans, Le processus de prospection informatique d'une Petite ou Moyenne Entreprise, FUNDP, Institut d'Informatique, Unité d'enseignement et de Recherche Architecture des Systèmes.
- [2] Apple Computer INC, Apple Pascal reference manual, Apple Computer INC, 10260 Bandley Drive, Cupertino, California 95014, 1979.
- [3] F. Bodart, Y. Pigneur, Conception assistée des applications informatiques, 1. Etude d'opportunité et analyse conceptuelle, collection méthodes + programmes, Masson, presses universitaires de Namur, 1983.
- [4] J. Colibri, Découvrez PASCAL sur Apple II, Mnemodyne S.A.R.L. 1980, 1980.
- [6] J.L. Hainaut, Conception assistée des applications informatiques, 2. Conception de la base de données. Masson, presses universitaires de Namur, 1986.
- [7] H. Haut, Gestion de fichiers et de périphériques pour Apple II/PASCAL (programmes utilitaires en PASCAL), édition du PSI, 1984.
- [8] R.J. Mc Quaker, Computer choice, a manual for the practitioner, North-Holland publishing Company - Amsterdam.New York.Oxford, 1978.
- [9] N. Magnenot-Thalman, D. Thalman, Gestion de fichiers et bases de données, deuxième édition, Gaëtan Marin éditeur.
- [10] Olivetti, Operating Systems and Languages Library UCSD p-system, Operating System User Guide, Olivetti, 1985.
- [11] Olivetti, Operating systems and Languages Library UCSD p-system, System Programmer's User Guide, Olivetti, 1985.
- [12] Olivetti, Operating Systems and Languages Library UCSD p-system, Program Development User Guide, Olivetti, 1985.
- [13] Parnas, Use of abstract interfaces in the development of software for embedded computer systems, NRL report 8047, 1977.

- [14] J.M. Poncelet, Savoir choisir et rentabiliser son informatique, Chambre de Commerce de Bruxelles, septembre 1982.
- [15] J. Thevenot, L'intégration des caractéristiques organisationnelles dans la conception du système d'information - propositions méthodologiques, thèse présentée pour l'obtention du doctorat ès science et gestion, 1985.
- [16] Tushman-Tadler, Information processing as an integrating concept in organizational design, the Academy of Management review vol 3 n° 3 july 1978.
- [17] R. Zaks, Votre ordinateur et Vous, SYBEX Europe, 1983.