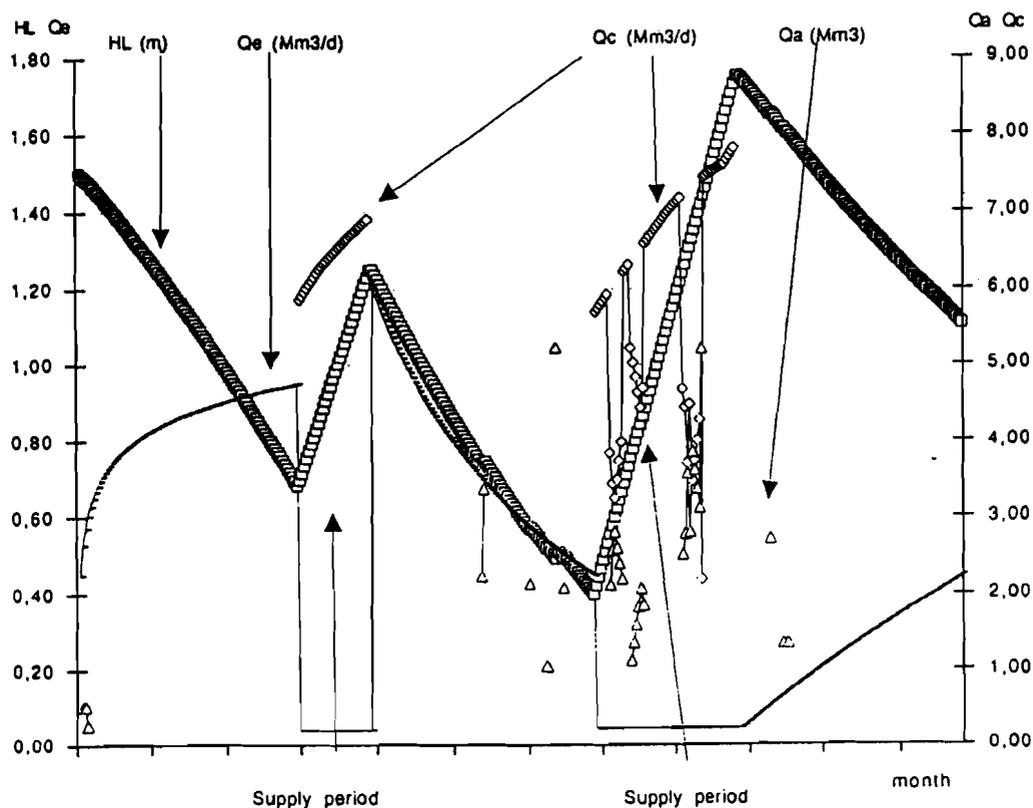


Noël Dacruz EVORA<sup>(1)</sup>, Jean Yves GAC<sup>(2)</sup>, François Xavier COGELS (1)

Projet CEE (EQUESEN) TS 2 0198 F EDB

## LOGICIEL DE GESTION PREVISIONNELLE (LGPLG) DES EAUX DU LAC DE GUIERS



-----  
Juin 1991

(1) Fondation Universitaire Luxembourgeoise (FUL), 185 avenue de Longwy, B- 6700, Arlon (Belgique)

(2) Institut Français de Recherche Scientifique pour le Développement en Coopération (ORSTOM), B.P. 1386, Dakar (Sénégal)

## INTRODUCTION

Le Logiciel de Gestion Prévisionnelle du Lac de Guiers (LGPLG) a été réalisé dans le but de :

- simuler la gestion quantitative de la réserve du lac sur la base de scénarios d'exploitation de la réserve fixés par les autorités compétentes.
- faciliter la prise de décision quant au choix d'une gestion optimale de la réserve lacustre.
- d'identifier plus précisément les contraintes de gestion additionnelles générées par un projet d'aménagement qui mobilisera une partie des ressources de la réserve lacustre.

Il pourra être utilisé par toutes les compétences techniques dont la gestion du lac est une préoccupation.

Le LGPLG a été écrit en langage Turbo pascal 5.0.

### I.- PRESENTATION GENERALE DU LGPLG

Le LGPLG comprend cinq (5) fonctions principales suivantes:

- 1) Création d'un fichier de données.
- 2) Modification des données d'un fichier.
- 3) Impression d'un fichier de données.
- 4) Calculs.
- 5) Graphiques.

Ces cinq fonctions apparaissent sur le menu principal.

L'appel d'une fonction particulière du LGPLG se fait par pression sur une touche numérique. Le caractère est lu sans "écho", il ne s'affiche pas à l'écran.

#### 1.- Création d'un fichier de données.

Cette fonction crée tous les fichiers nécessaires aux calculs. Le fichier créé est un fichier typé. Chaque composant du fichier est isolé par rapport aux autres. Il est directement accessible par l'identification de sa position au sein du fichier.

Les informations fournies sur les différents fichiers sont assez explicites pour guider l'utilisateur du logiciel.

Les fichiers sont composés de données journalières. Elles peuvent être introduites quotidiennement ou par tranches de jours.

En annexe figure la liste exhaustive des fichiers pouvant être créés ainsi qu'une description succincte de chaque fichier.

## **2.- Modification des données d'un fichier.**

Les sous-fonctions permettent de corriger les données erronées d'un fichier, d'ajouter ou de supprimer des données sans qu'il soit nécessaire de recréer un nouveau fichier.

### *2.1 Ajout de données à un fichier.*

Le logiciel demande le nombre de données à ajouter et le rang après lequel les nouvelles données seront introduites.

N.B. : Pour ajouter un élément en début de fichier, le rang doit valoir 0.

### *2.2 Suppression de données d'un fichier.*

Il est possible de supprimer le nombre de données voulu à partir du rang spécifié.

## **3.- Impression d'un fichier.**

Tous les fichiers de données et de résultats peuvent être imprimés. Pour un fichier contenant plus de 509 données, l'impression doit commencer en début de page, le saut de page étant géré par le logiciel.

Pour imprimer un fichier, il faut introduire le nom du fichier à imprimer, un titre général faisant référence au fichier et l'unité relative aux données contenues dans le fichier. Le titre général et l'unité sont des informations facultatives.

#### 4.- Calculs.

Cette fonction permet de réaliser les tâches de calcul suivantes:

- a) Cote du lac à la fin de la période de simulation.
- b) Cote du lac au début de la période de simulation.
- c) Réajustement du type d'évolution de la cote du lac.
- d) Apport fluvial et transfert vers Diama.
- e) Besoins en eau d'irrigation.

Les calculs pour les cinq sous-fonctions proposées se font au pas de temps journalier.

##### 4.1 *Calcul de la cote du lac à la fin de la période de simulation.*

La cote à la fin de la période de simulation est calculée connaissant la cote initiale et l'ensemble des apports en eau à la réserve et des pompes sollicitant cette même réserve.

Les données journalières sur les apports et les pertes en eau figurent dans des fichiers qui auront été créés préalablement (fichiers APPORTS et PERTES).

Les résultats sont inscrits dans les fichiers figurant dans le groupe AUTRES FICHIERS. Ces fichiers sont créés au moment du calcul et sont énumérés ci-dessous:

##### . *Cote du lac:*

Dans ce fichier se trouvent les cotes journalières du lac calculées durant la période de simulation.

##### . *Cote au quai de Richard-Toll:*

Durant la phase de remplissage du lac par les eaux du fleuve Sénégal, la cote requise au quai de Richard-Toll pour assurer l'apport en eau désiré est calculée et inscrite dans ce fichier.

##### . *Volume évaporé quotidiennement:*

Dans ce fichier figurent les volumes d'eau évaporés quotidiennement durant la période de simulation.

Remarque:

Pour les quatre premières sous-fonctions du menu Calculs, les fichiers apports et les fichiers pertes sont identiques.

FICHIERS APPORTS

Fleuve  
Rejets CSS au lac  
Précipitations

FICHIERS PERTES

Pompages CSS  
Pompages SONEES  
Pompages Canal du CAYOR  
Pompages Irrigation  
Pompages Supplémentaires  
Evaporation journalière

Le fichier *pompages IRRIGATION* est le résultat du calcul réalisé par la sous-fonction 5 du menu Calculs.

Le fichier *pompages Supplémentaires* sert à introduire tous les autres pompages effectués dans la réserve lacustre et qui ne figurent pas expressément dans la liste des fichiers PERTES. Il peut s'agir d'un transfert d'eau vers la réserve de Diama ou vers le Ferlo lors de l'ouverture des digues de Keur Momar Sarr, etc...

Le fichier *Evaporation journalière* est constitué par les valeurs moyennes journalières de l'évaporation du lac de Guiers. Ces valeurs quotidiennes ont été déterminées à partir des données moyennes mensuelles de l'évaporation établies sur la période 1976-1989. A chacun des jours de l'année, il a été attribué une évaporation-type (Cogels et al., 1991).

Les calculs sont effectués après avoir introduit le nombre de jours de simulation, le code du jour de début de période de simulation et la cote du lac au début de la période de simulation (cote initiale). Cette cote est celle du début du premier jour de la période de simulation.

A l'issue des calculs, la cote à la fin de la période de simulation (cote finale) est affichée ainsi que le bilan hydrologique en m sur la période de simulation.

Lorsque la simulation est faite sur n jours, les fichiers APPORTS et PERTES doivent contenir chacun au minimum n données. Dans le cas contraire, une erreur en lecture sur la disquette ou le disque dur se produit à la fin de la simulation (erreur 100). Le calcul terminé, le fichier *Cote du lac* (AUTRES FICHIERS) contiendra n+1 données. La dernière donnée de ce fichier est la cote du lac à la fin du jour n ou au début du jour n+1.

#### 4.2 *Calcul de la cote du lac au début de la période de simulation.*

Cette fonction réalise l'opération inverse de celle décrite précédemment. Elle permet de calculer la cote du lac au début d'une période de simulation à partir de la cote connue en fin de période de simulation, introduite comme donnée, et de l'ensemble des apports et pertes en eau du lac.

Les calculs sont effectués après avoir introduit le nombre de jours de simulation, le code du jour de fin de période de simulation et la cote du lac à la fin du jour de fin de période de simulation (cote finale).

Les données des fichiers du groupe AUTRES FICHIERS sont d'abord remises dans l'ordre croissant du temps avant que la cote du lac calculée au début de la période de simulation ne soit affichée. Cette cote calculée (cote initiale) correspond à la cote du lac au début du premier jour de la période de simulation. Cette opération de remise en ordre des fichiers prépare ces derniers à la visualisation graphique. Le bilan hydrologique en m est présenté ensuite.

#### 4.3 *Réajustement du type d'évolution de la cote du lac.*

Cette troisième possibilité de calcul permet de savoir s'il y a une adéquation entre le choix d'une évolution de la cote du lac et l'ensemble des pompages prévus et des apports en eau escomptés.

Outre les fichiers de données usuels, il y a les fichiers *Evolution choisie de la cote du lac* et *Remplissage ou non du lac*. Le premier fichier contient les cotes journalières du lac fixées par le gestionnaire et le deuxième est composé de 0 et de 1. Pour un jour quelconque, la valeur 1 signifie que le lac est en phase de remplissage (communication fleuve-lac) et la valeur 0 que le lac évolue en système fermé.

La première donnée du fichier *Evolution choisie de la cote du lac* est considérée comme la cote du lac au début de la période de simulation. A partir de cette donnée, le calcul s'effectue comme pour la sous-fonction 1 du menu Calculs. Les cotes journalières calculées s'inscrivent dans le fichier *Cotes du lac réajustées*. Les apports fluviaux (fichier *fleuve*) sont calculés à partir des cotes du fichier *Evolution choisie de la cote du lac*.

Lorsque le lac évolue en système fermé, le logiciel calcule un apport en eau fictif du fleuve ou au contraire un transfert fictif d'eau vers la réserve de Diama nécessaire au maintien des cotes du lac au niveau fixé préalablement par le gestionnaire (fichier *Evolution choisie de la cote du lac*).

En phase de remplissage, le logiciel évalue les apports du fleuve à partir des cotes fixées par le gestionnaire. Pendant cette phase, les cotes réajustées sont calculées en considérant des apports du fleuve identiques.

En dehors des fichiers de données, le nombre de jours de simulation et le code du jour de début de période de simulation sont fournis au logiciel.

Les résultats du calcul sont écrits dans le fichier *Fleuve* (FICHIERS APPORTS) et le fichier *Cotes du lac réajustées* (AUTRES FICHIERS).

Quand la simulation est faite sur n jours, le fichier *Evolution choisie de la cote du lac* doit contenir n+1 données. La dernière valeur de ce fichier correspond à la cote à la fin du jour n ou à celle au début du jour n+1. De même, le fichier *Cotes du lac réajustées* a n+1 composants.

#### 4.4 *Apport fluvial et transfert vers Diama.*

Cette fonction est utile au calcul de l'apport fluvial et du transfert d'eau vers Diama à partir de cotes du lac connues (fichier *Cote du lac*) et de la définition des périodes de communication fleuve-lac (fichier *Remplissage ou non du lac*).

Elle peut également servir à affiner le profil de l'apport fluvial qui sera utilisé dans d'autres calculs.

Le nombre de jours de simulation ainsi que le code du jour de début de période de simulation doivent être fournies au logiciel.

Les résultats du calcul s'inscrivent dans le fichier *Fleuve* (FICHIERS APPORTS) et le fichier *Volume évaporé quotidiennement* (AUTRES FICHIERS). Le logiciel donne aussi une évaluation de l'apport fluvial total et du transfert d'eau total vers Diama durant la période de simulation.

En effectuant un calcul sur n jours, tous les fichiers de données doivent contenir au minimum n données.

#### 4.5 *Calcul des besoins en eau d'irrigation.*

Les spéculations agricoles sont les suivantes:

- 1) Canne à sucre.
- 2) Riz d'hivernage.
- 3) Riz de contre-saison.
- 4) Maïs d'hivernage.

- 5) Maïs de contre-saison.
- 6) Légumes.
- 7) Tomates.
- 8) Divers.

Le poste divers offre la possibilité de tenir compte d'une culture non spécifiée par le logiciel.

Les données à fournir au logiciel sont le nombre de jours de calcul, le nom du fichier où seront inscrits les besoins en eau d'irrigation et ensuite, au fur et à mesure, pour chaque spéculation agricole, les fichiers surfaces cultivées en hectares, les fichiers besoins en eau en m<sup>3</sup> par hectare et par jour ainsi que les fichiers efficience de l'irrigation en pourcentage.

Les besoins totaux en eau pour l'irrigation sont exprimés en millions de m<sup>3</sup>.

## 5.- Graphiques

Les graphiques sont de deux types, courbe ou histogramme.

### 5.1 Courbe.

Pour ce type de graphique, il faut introduire le nom du fichier qui sera visualisé, puis le titre du graphique (maximum de 60 caractères) et le titre de l'unité de l'axe des y (maximum de 20 caractères). Le titre du graphique et l'unité de l'axe y sont facultatifs.

Il n'est pas possible de visualiser un fichier composé de valeurs journalières constantes. Une erreur survient alors (division par zéro).

### 5.2 Histogramme.

Ce type de graphique permet de visualiser uniquement le bilan hydrologique du lac exprimé en m et seulement après avoir calculé la cote du lac à la fin de la période de simulation ou la cote du lac au début de la période de simulation (sous-fonctions 1 ou 2 du menu Calculs). Tant qu'une autre simulation n'a pas été entamée, il est possible de visualiser l'histogramme puisque les valeurs restent en mémoire.

Un titre peut être donné au graphique (maximum de 60 caractères). Il est facultatif.

## II.- CONSIDERATIONS PARTICULIERES

### 1.- Constitution d'un dossier pour chaque simulation

Il est intéressant de disposer pour chaque simulation effectuée des informations essentielles (fichiers de données et de résultats, nombre de jours de simulation, etc...) pour un rappel ultérieur. Pour passer d'un écran à un autre, il faut nécessairement appuyer sur la touche Retour. L'écran peut alors être imprimé grâce à la touche Print Screen (Prt Sc) avant de presser sur la touche Retour.

### 2.- Nombre de jours de simulation

Théoriquement, le nombre de jours de simulation est illimité puisqu'il est entièrement défini par l'opérateur. Cependant, cette donnée a été limitée à 730 ou 731 jours soit deux années de simulation. Pour ce faire, trois fichiers *Evaporation journalière* ont été déjà créés. Il s'agit de:

*evapjmb.dat* : il contient 730 données sur deux années non bissextiles.

*evapjmb1.dat* : il contient 731 données sur deux années dont la première est bissextile.

*evapjmb2.dat* : il contient 731 données sur deux années dont la deuxième est bissextile.

Le code des jours varie de 1 à 730 ou 731 selon que les deux années sont non bissextiles ou que l'une d'elles est bissextile. Le code 1 correspond au 1<sup>er</sup> janvier et le code 730 ou 731 au 31 décembre de la deuxième année. Une simulation peut être commencée à n'importe quel moment de l'année.

Il ne faut pas confondre le code des jours évoqué ci-dessus avec le code de référence des données d'un fichier lorsque celui-ci a été imprimé. Ces deux codes sont équivalents dans le cas où la simulation débute le 1<sup>er</sup> janvier.

### 3.- Calcul de la hauteur au quai de Richard-Toll

C'est une information complémentaire donnée par le logiciel qui calcule la hauteur d'eau qu'il faut assurer au quai de Richard-Toll pour avoir un débit  $Q$  connu de l'apport fluvial.

La campagne de jaugeages sur le canal de la Taoué (Lamagat et al., 1990) pendant l'hivernage 1990 a permis d'établir les équations suivantes qui évalue le débit  $Q$  associé à une hauteur  $h$  et à une dénivelée  $D$  :

- la courbe de tarage à dénivelée normale  $D_n$

$$Q_n = f(h, D_n) \quad (1)$$

- la courbe de correction du débit

$$Q_n = Q \cdot (D_n / (D + dh))^n \quad (2)$$

$Q_n$  : Débit correspondant à la dénivelée normale  $D_n$ .

$dh$  : Erreur de calage entre les deux échelles.

$D$  : dénivelée  $h_{\text{fleuve}} - h_{\text{quai}}$ .

Les paramètres  $n$ ,  $D_n$  et  $dh$  valent respectivement 0.57, 8 et 5. Ainsi  $Q_n$  ne dépend que de la hauteur d'eau à l'échelle du quai de Richard-Toll.

La courbe  $Q_n = f(h_{\text{quai}})$  a été discrétisé de la façon suivante:

$60\text{cm} \leq h_{\text{quai}} \leq 80\text{cm}$	$Q_n = 0.01h + 23.4$	
$80\text{cm} \leq h_{\text{quai}} \leq 100\text{cm}$	$Q_n = 0.11h + 15.4$	
$100\text{cm} \leq h_{\text{quai}} \leq 120\text{cm}$	$Q_n = 0.36h - 9.6$	(3)
$120\text{cm} \leq h_{\text{quai}} \leq 140\text{cm}$	$Q_n = 0.50h - 26.4$	
$140\text{cm} \leq h_{\text{quai}} \leq 160\text{cm}$	$Q_n = 0.64h - 46.0$	

L'approximation consistant à considérer que la hauteur du plan d'eau à Ngnith et la hauteur du lac à l'embouchure de la Taoué étaient identiques a été faite.

Le calcul de  $h_{\text{quai}}$  se fait par itérations successives jusqu'à ce que la différence des  $Q_n$  pour les équations (1) et (2) soit faible.

$Q_n$  (3) n'est déterminé que pour des hauteurs au quai de Richard-Toll comprise entre 60 et 160 cm. Ainsi, lorsque pour une hauteur du lac donnée,  $h_{\text{quai}}$  n'est pas trouvé dans la gamme de hauteurs précédente, le processus itératif s'arrête pour  $h_{\text{quai}} = 161$  cm. Si le débit  $Q$  est nul alors  $h_{\text{quai}}$  vaut 0.

#### 4.- Installation de l'environnement graphique.

Cette installation est nécessaire pour pouvoir visualiser les graphiques. Le logiciel installe automatiquement le pilote graphique et définit le mode graphique.

Si une erreur se produit au moment de l'installation de l'environnement graphique, le code de l'erreur est affiché ainsi que le message correspondant.

La signification des messages en fonction des codes d'erreur possibles est donné en annexe.

#### 5.- Gestion des erreurs à l'ouverture de fichier

Pour éviter des interruptions du logiciel suite à des erreurs produites au moment de l'ouverture d'un fichier lorsque celui-ci est non disponible (chemin d'accès erroné) ou inexistant (fichier non créé), la vérification automatique d'erreur est inhibée avant l'ouverture du fichier puis rétablie immédiatement après.

Si une erreur se produit à l'ouverture d'un fichier, un message est envoyé à l'utilisateur. Après avoir appuyé sur la touche Retour, l'opération précédente peut alors être reprise à nouveau.

Pour les sous-fonctions du menu Calculs, il faut introduire un nouveau nom de fichier à chaque fois qu'il y a une erreur à l'ouverture d'un fichier. Le cas échéant, le curseur revient à sa position initiale après avoir effacé le mauvais nom de fichier.

#### 6.- Installation du logiciel sur le disque dur

- . Créer sur le disque dur un répertoire du nom de LGPLG.
- . Copier tous les fichiers .BGI et .CHR de la disquette dans ce répertoire.
- . Le logiciel est lancé en tapant lgplg et Retour.

#### 7.- Quelques exemples d'application

- la hauteur minimale de remplissage du lac qui permet de minimiser l'évaporation et de ne pas dépasser au bout d'un nombre de jours fixé un niveau de cote du lac choisi par le gestionnaire de la réserve lacustre peut être calculée avec la sous-fonction 2 du menu Calculs.
- En fonction d'un programme de prélèvements dans la réserve lacustre et des apports en eau escomptés, il est possible de connaître la cote du lac en fin d'année en considérant la cote connue au 1<sup>er</sup> janvier.

- Si on désire ne pas dépasser une cote précise en fin d'année ou à n'importe quel moment de l'année, la sous-fonction 2 du menu Calculs doit être employée.
-

**REFERENCES BIBLIOGRAPHIQUES**

COGELS F.X., DACRUZ EVORA N., GAC J.Y. (1991) - L'évaporation du lac de Guiers (Sénégal) de 1976 à 1989. Bilan et essai d'interprétation. Rapport CEE, Projet EQUÉSEN (TS 2 0198 F EDB) et Rapport ORSTOM, Dakar, 25 p.

LAMAGAT J.-P., SEGUIS L., DUBEE G., PEPIN Y. (1990) - Campagne de jaugeage sur le canal de la Taoué (Hivernage 1990). Rapport OMVS, Projet "Prévision des débits de gestion des ouvrages communs de l'OMVS" et Rapport ORSTOM, Dakar, 11 p.

ROSENBAUM G., SCHÖLLES R. (1989) - Le grand livre du Turbo Pascal. Editions MICRO APPLICATION, 58, rue du Faubourg Poissonnière, 75010 PARIS, 669 p.

---

**A N N E X E S**

---

## FICHIERS DE DONNEES CREES PAR LE LGPLG

- . *Rejets CSS au lac*  
Volume d'eau de drainage des casiers sucriers de la CSS (Compagnie Sucrière Sénégalaise) et déversé dans le lac.
- . *Précipitations*  
Lame d'eau moyenne reçue par le lac.
- . *Apports du fleuve*  
Volume d'eau en provenance du fleuve et transitant par le canal de la Taoué vers le lac (crue fluviale ou réserve de Diama).
- . *Pompages CSS*  
Volume d'eau pompé par la CSS pour l'irrigation des casiers sucriers.
- . *Pompages SONEES*  
Volume d'eau pompé par la SONEES (Société Nationale d'Exploitation des Eaux du Sénégal) à Ngnith pour l'alimentation en eau potable.
- . *Pompages Canal du Cayor*  
Volume d'eau transitant par le futur canal du Cayor.
- . *Pompages supplémentaires*  
Tous pompages ou soutirages d'eau effectués dans la réserve lacustre qui ne seraient pas pris en compte par les pompages mentionnés ci-dessous. Il peut s'agir d'un transfert d'eau de la réserve lacustre vers celle de Diama ou vers la vallée fossile du Ferlo, etc...
- . *Evaporation journalière*  
Lame d'eau moyenne évaporée quotidiennement au lac.
- . *Cote du lac*  
Cotes journalière du lac à Ngnith.

. *Remplissage ou non du lac*

Uniquement composé de 0 et de 1, ce fichier permet de spécifier toutes les périodes durant lesquelles le lac est ou non en communication avec le fleuve.

0 : le lac est isolé.

1 : le lac communique avec le fleuve.

. *Surfaces cultivées*

Surfaces cultivées en hectare pour chaque spéculation agricole.

. *Besoins en eau*

Volume d'eau d'irrigation en m<sup>3</sup> par hectare pour chaque spéculation agricole.

. *Efficiencce de l'irrigation*

Exprimé en pourcentage pour chaque spéculation agricole (valeur comprise entre 1 et 100).

## INSTALLATION DE L'ENVIRONNEMENT GRAPHIQUE

### SIGNIFICATION DES MESSAGES D'ERREURS

Code	Message	Signification
0	grOk	Exécution sans erreur.
-1	grNoInitGraph	Pilote graphique non installé.
-2	grNotDetected	Aucun adaptateur graphique n'est disponible/le mode demandé ne peut être installé.
-3	grFileNotFound	Fichier de pilote graphique introuvable.
-4	grInvalidDriver	Programme de pilote graphique défectueux.
-5	grNoLoadMem	Mémoire de travail insuffisante pour le chargement du pilote graphique.
-6	grNoScanMem	Mémoire de travail insuffisante (ScanFill).
-7	grNoFloodMem	Mémoire de travail insuffisante (FloodFill).
-8	grFontNoFound	Fichier de police introuvable.
-9	grNoFontMem	Mémoire de travail insuffisante pour charger la police de caractères.
-10	grInvalidMode	Le pilote chargé ne supporte pas le mode graphique demandé.
-11	grError	Erreur générique.
-12	grIOerror	Erreur d'entrée/sortie dans le programme graphique.
-13	grInvalidFont	Fichier de police défectueux.
-14	grInvalidFontNum	Numéro de police erroné.
-15	grInvalidDeviceNum	Chemin d'accès à la police erroné.

=====

```

{-----}
{ unit      : lgplgcb.pas      }
{-----}
{ date      : 19/07/91        }
{ compiler  : turbo pascal 5.0 }
{-----}
{ Auteur   : Noël Dacruz BVORA }
{-----}

```

```

{ Cette unité visualise sur écran graphique }
{ l'évolution temporelle des données d'un fichier }
{ typé.                                     }
{-----}

```

```
unit lgplgcb;
```

```

{-----}
{      Interface      }
{-----}

```

```
Interface
uses crt, graph, printer;
```

```

var
i, k, long, abscisse, transfert, val_err :integer;
res, max, min, element, valeur :real;
x, x1, x2, x3, x4, y, y1, y2, y3, y4, y5, y6, y7 :longint;
part_chaine :string[3];
chaine :string[25];
titre_y :string[20];
nom :string[30];
titre :string[60];
f :file of real;

```

```

procedure tri;
procedure position_abs;
function etiquette (xr :real) :string;
procedure constant;
procedure fond (G_titre :string);
procedure stylaxe;
procedure axe_x;
procedure axe_y (tit_y: string);
procedure graphiq_pixel;
procedure courbe (nom1_titre, nom1_titre_y :string);

```

```

{-----}
{      Implementation      }
{-----}

```

```
Implementation
```

```

{-----}
{      tri      }
{      trouve les valeurs maximum et }
{      minimum d'un fichier          }
{-----}

```

```

procedure tri;
begin
reset(f);
read(f,max); read(f,min);
if max<min

```

Nom du programme : b:lgplgcb.pas

Heure : 12:06

```

=====

then begin
    element:=min;
    min:=max;
    max:=element;
end;
for i:=1 to filesize(f)-2 do
begin
    read(f,element);
    if element>max then max:=element;
    if element<min then min:=element;
end;
end;
{-----}
{      position_abs      }
{      définit la position des }
{      étiquettes de l'axe X   }
{-----}
procedure position_abs;
begin
    long:=length(chaine);
    case long of
        1 : abscisse:=58;
        2 : abscisse:=56;
        3 : abscisse:=53;
    end; { case }
end;
{-----}
{      fonction etiquette      }
{      définition des étiquettes }
{      de l'axe Y              }
{-----}
function etiquette (xr :real) :string;
begin
    str(xr,chaine);
    part_chaine:=copy(chaine,15,3);
    val(part_chaine,transfert,val_err);
    if transfert<0
    then begin
        delete(chaine,3,1);
        for i:=1 to abs(transfert) do insert('0',chaine,2);
        insert('.',chaine,3);
    end
    else if transfert>0
    then begin
        delete(chaine,3,1);
        insert('.',chaine,transfert+3);
    end;
    etiquette:=copy(chaine,1,6);
end;
{-----}
{      constant      }
{-----}
procedure constant;
begin
    x:=getmaxx-40; x1:=getmaxx-35; x2:=getmaxx-100; x3:=getmaxx-42;
    y:=getmaxy-40; y1:=getmaxy-42; y2:=getmaxy-38; y3:=getmaxy-43;
    y4:=getmaxy-45; y5:=getmaxy-50; y6:=getmaxy-110; y7:=getmaxy-30;
end;

```

Nom du programme : b:lgplgcb.pas

Heure : 12:08

```

=====
{-----}
{      fond      }
{-----}
procedure fond (G_titre :string);
begin
  setbkcolor(blue);
  settxtstyle(0,0,1);
  outtextxy(20,10,'GRAPHIQUE:');
  outtextxy(115,10,G_titre);
  setlinestyle(0,0,3);
  setcolor(yellow);
  rectangle(10,30,getmaxx-10,getmaxy-10);
end;
{-----}
{      stylaxe   }
{-----}
procedure stylaxe;
begin
  setcolor(white);
  setlinestyle(0,0,1);
  settxtstyle(2,0,4);
end;
{-----}
{      axe_x     }
{      trace l'axe X   }
{-----}
procedure axe_x;
begin
  line(60,y,x,y); line(x1,y,x,y1);
  line(x,y,x,y2); line(x,y1,x1,y);
  line(x,y2,x1,y);
  i:=1;
  outtextxy(60,y3,'');
  str(i,chaine);
  outtextxy(59,getmaxy-30,chaine);
  repeat
    i:=i+round(filesize(f)/6);
    outtextxy(60+round((i-1)*abs(x2/(filesize(f)-1))),y3,'');
    str(i,chaine); position_abs;
    outtextxy(abscisse+round((i-1)*abs(x2/(filesize(f)-1))),y7,chaine);
  until i>filesize(f)-round(filesize(f)/6);
  outtextxy(x3,y7,'jours');
end;
{-----}
{      axe_y     }
{      trace l'axe Y   }
{-----}
procedure axe_y (tit_y: string);
begin
  line(60,60,60,y); line(60,60,62,60); line(60,60,58,60);
  line(58,60,60,55); line(62,60,60,55);
  outtextxy(25,40,tit_y);
  outtextxy(55,y4,'-');
  valeur:=min;
  outtextxy(15,y4,etiquette(valeur));
  for k:=1 to 2 do
  begin
    outtextxy(55,y5-round(y6*k/3),'-');
  end;
end;

```

Nom du programme : b:lgplgcb.pas

Heure : 12:10

=====

```

    valeur:=min+(k*(max-min)/3);
    outtextxy(15,y5-round(y6*k/3),etiquette(valeur));
end;
outtextxy(55,65,'-');
valeur:=max;
outtextxy(15,65,etiquette(valeur));
end;
{-----}
{      graphiq_pixel      }
{      réalise un graphique par pixels      }
{-----}
procedure graphiq_pixel;
begin
  reset(f);
  read(f,element);
  res:=element-min;
  setcolor(lightcyan);
  moveto(60,(y-round((res*y6/(Max-min)))));
  for i:=2 to filesize(f) do
  begin
    read(f,element);
    res:=element-min;
    lineto(60+round(((i-1)*abs(x2/(filesize(f)-1)))),
            (y-round((res*y6/(Max-min)))));
  end;
  readln;
  close(f);
  clonegraph;
end;
{-----}
{      courbe      }
{-----}
procedure courbe (noml_titre, noml_titre_y :string);
begin
  tri;
  constant;
  fond(noml_titre);
  stylaxe;
  axe_x;
  axe_y(noml_titre_y);
  graphiq_pixel;
end;
end.

```

Nom du programme : b:lgplghg.pas

Heure : 11:51

=====

```

{-----}
{ unit   : lgplghg.pas }
{-----}
{ date   : 10/07/91   }
{ compiler : turbo pascal 5.0 }
{-----}
{ Auteur  : Noël Dacruz EVORA }
{-----}

```

```

{ Cette unité visualise sur écran graphique }
{ et sous forme d'histogramme les composantes }
{ du bilan hydrologique exprimées en m d'eau. }
{-----}

```

```
unit lgplghg;
```

```

{-----}
{      Interface      }
{-----}

```

```
Interface
```

```
uses crt, graph, lgplgcb;
```

```
type
```

```
  indice =array[1..9] of real;
```

```
const
```

```
  grad:array[1..9] of string[6]=('PLUIE','REJETS','PLEUVE','CSS','SONEBS',
                                'CAYOR','IRRIG.','SUPPL.','EVAP.');
```

```
var
```

```
poste :indice;
```

```
procedure max_n;
```

```
procedure axe_x_h;
```

```
procedure axe_y_h;
```

```
procedure legende;
```

```
procedure histogramme;
```

```

{-----}
{      Implementation      }
{-----}

```

```
Implementation
```

```

{-----}
{      max_n               }
{      trouve la valeur maximum }
{-----}

```

```
procedure max_n;
```

```
begin
```

```
  max:=poste[1];
```

```
  for i:=2 to 9 do if poste[i]>max then max:=poste[i];
```

```
end;
```

```

{-----}
{      axe_x_h             }
{      trace l'axe X de l'histogramme }
{-----}

```

```
procedure axe_x_h;
```

```
begin
```

```
  line(60,y,x,y); line(x1,y,x,y1);
```

Nom du programme : b:lgplghg.pas

Heure : 11:52

=====

```

line(x,y,x,y2); line(x,y1,x1,y);
line(x,y2,x1,y);
x4:=round((getmaxx-110)/9);
for i:=1 to 10 do outtextxy(60+(i-1)*x4,y3, '|');
outtextxy(60+round(x4/4), y3+15, grad[1]);
outtextxy(60+round(x4+x4/4.5), y3+15, grad[2]);
outtextxy(60+round(2*x4+x4/4.5), y3+15, grad[3]);
outtextxy(60+round(3*x4+x4/2.8), y3+15, grad[4]);
outtextxy(60+round(4*x4+x4/4.5), y3+15, grad[5]);
outtextxy(60+round(5*x4+x4/4), y3+15, grad[6]);
outtextxy(60+round(6*x4+x4/4), y3+15, grad[7]);
outtextxy(60+round(7*x4+x4/4), y3+15, grad[8]);
outtextxy(60+round(8*x4+x4/3.5), y3+15, grad[9]);
end;
{-----}
{      axe_y_h      }
{      trace l'axe Y de l'histogramme      }
{-----}
procedure axe_y_h;
begin
line(60,60,60,y); line(60,60,62,60); line(60,60,58,60);
line(58,60,60,55); line(62,60,60,55);
outtextxy(25,40,'m d'eau');
outtextxy(55,y4,'-');
outtextxy(21,y5+5,'0');
for k:=1 to 2 do
begin
outtextxy(55,y5-round(y6*k/3),'-');
valeur:=k*max/3;
outtextxy(15,y5-round(y6*k/3),etiquette(valeur));
end;
outtextxy(55,65,'-');
valeur:=max;
outtextxy(15,65,etiquette(valeur));
end;
{-----}
{      legende      }
{      trace la légende      }
{-----}
procedure legende;
begin
setfillstyle(1,2);
bar(150,45,170,55);
rectangle(150,45,170,55);
outtextxy(180,45,'APPORTS');
setfillstyle(1,3);
bar(300,45,320,55);
rectangle(300,45,320,55);
outtextxy(330,45,'PERTES');
end;
{-----}
{      histogramme      }
{      trace un histogramme      }
{-----}
procedure histogramme;
begin
setcolor(white);
for i:=1 to 9 do

```

=====

```
if i in [1..3]
then begin
  setfillstyle(1,2);
  bar(60+round((i-1)*x4),y-round(poste[i]*y6/max),60+round(i*x4),y);
  rectangle(60+round((i-1)*x4),y-round(poste[i]*y6/max),60+round(i*x4),y);
end
else begin
  setfillstyle(1,3);
  bar(60+round((i-1)*x4),y-round(poste[i]*y6/max),60+round(i*x4),y);
  rectangle(60+round((i-1)*x4),y-round(poste[i]*y6/max),60+round(i*x4),y);
end;
readln;
closegraph;
end;
end.
```

=====

```

{-----}
{ unit      : gestfich.pas }
{ }
{ date      : 10/07/91 }
{ compiler  : turbo pascal 5.0 }
{ }
{ Auteur   : Noël Dacruz EVORA }
{ }

```

```

{-----}
{ Cette unité permet de créer un fichier de }
{ données, de corriger les données erronées, }
{ de supprimer des données ou d'en ajouter à }
{ un fichier. }
{ Il imprime aussi le fichier de données. }
{ }
{ Remarque: Le fichier est un fichier typé. }
{-----}

```

```

unit gestfich;
{-----}
{      Interface      }
{-----}

```

```

Interface
uses crt, printer, t_io;

var
  res :word;
  un  :string[20];
  ch  :char;
  nom :string[30];
  rmc :string[41];
  titre :string[60];
  i , j, k, dt, nbre_jours, ft, ntvc, nbr_don, rang :integer;
  element, element1 :real;
  f :file of real;

```

```

procedure message;
procedure entetel;
procedure introduction1;
procedure introduction2;
procedure menu_creat_fich;
procedure affiche;
procedure autr_fich;
procedure c_menu_creat_fich;
procedure creat_fich;
procedure correct_fich (nom1 :string);
procedure ajout_fich (nom1 :string);
procedure suppr_fich (nom1 :string);
procedure menu_mod_fich;
procedure c_menu_mod_fich;
procedure impr_fich;

```

```

{-----}
{      Implementation      }
{-----}

```

```

Implementation

{-----}
{      message      }
{-----}

```

```
=====

{ En cas d'erreur à l'ouverture d'un fichier }
{-----}
procedure message;
begin
  textcolor(lightgreen);
  t_ecrit(48,19,' '); t_horizontal(49,19,24,196); t_ecrit(73,19,' ');
  t_vertical(48,20,3,179); t_vertical(73,20,3,179);
  t_ecrit(48,23,'L'); t_horizontal(49,23,24,196); t_ecrit(73,23,'J');
  gotoxy(50,20); write('Fichier non disponible');
  gotoxy(60,21); write('ou');
  gotoxy(52,22); write('Fichier inexistant');
  readln;
end;
{-----}
{ entetel }
{ procedure de creation_fichier }
{-----}
procedure entetel;
begin
  gotoxy(6,7); write('TRANCHE ',j:2);
  gotoxy(22,9); write('Jour de début : '); readln (dt);
  gotoxy(22,11); write('Jour de fin : '); readln (ft);
end;
{-----}
{ introduction1 }
{ procedure de creation_fichier }
{-----}
procedure introduction1;
begin
  gotoxy(19,7); writeln('Introduisez vos données');
  for i:=(filesize(f)+1) to (dt-1) do
  begin
    gotoxy(18,11); write('valeur n',i:4,' ..... ');
    gotoxy(44,11); clrcol; readln(element);
    write(f,element);
  end;
end;
{-----}
{ introduction2 }
{ procedure de creation_fichier }
{-----}
procedure introduction2;
begin
  gotoxy(13,10);
  write('Valeur constante de la tranche ', j:2, ' : ');
  readln(element);
  for i:=dt to ft do write(f,element);
end;
{-----}
{ affiche }
{ affiche les spéculations agricoles }
{-----}
procedure affiche;
begin
  window(1,8,80,25); clrscr; window(1,1,80,25);
  gotoxy(30,6); write(rmc);
  gotoxy(30,10); write('1. Canne à sucre');
  gotoxy(30,11); write('2. Riz hivernage');
```

Nom du programme : b:gestfich.pas

Heure : 12:21

```

=====

gotoxy(30,12); write('3. Riz contre-saison');
gotoxy(30,13); write('4. Maïs hivernage');
gotoxy(30,14); write('5. Maïs contre-saison');
gotoxy(30,15); write('6. Légumes');
gotoxy(30,16); write('7. Tomates');
gotoxy(30,17); write('8. Divers');
gotoxy(34,21); write('Votre choix ? ');
repeat
  ch:=readkey;
until ch in ['1'..'8'];
case ch of
  '1' :begin
    rmc:=rmc+' CANNE A SUCRE'; end;
  '2' :begin
    rmc:=rmc+' RIZ HIVERNAGE'; end;
  '3' :begin
    rmc:=rmc+' RIZ CONTRE-SAISON'; end;
  '4' :begin
    rmc:=rmc+' MAIS HIVERNAGE'; end;
  '5' :begin
    rmc:=rmc+' MAIS CONTRE-SAISON'; end;
  '6' :begin
    rmc:=rmc+' LEGUMES'; end;
  '7' :begin
    rmc:=rmc+' TOMATES'; end;
  '8' :begin
    rmc:=rmc+' DIVERS'; end;
end;
end;
{-----}
{   menu_creat_fich   }
{-----}
procedure menu_creat_fich;
begin
  clrscr; textcolor(lightcyan);
  gotoxy(25,2); write('CREATION D'UN FICHIER DE DONNEES');
  gotoxy(25,3); write('-----');
  gotoxy(30,8); write('1. Rejets CSS au lac');
  gotoxy(30,9); write('2. Précipitations');
  gotoxy(30,10); write('3. Apports du fleuve');
  gotoxy(30,11); write('4. Pompages CSS');
  gotoxy(30,12); write('5. Pompages SONEES');
  gotoxy(30,13); write('6. Pompages Canal du Cayor');
  gotoxy(30,14); write('7. Pompages supplémentaires');
  gotoxy(30,15); write('8. Evaporation journalière');
  gotoxy(30,16); write('9. Autres fichiers'); textcolor(yellow);
  gotoxy(30,17); write('0. Menu Précédent'); textcolor(lightcyan);
  gotoxy(34,21); write('Votre choix ? ');
end;
{-----}
{   autr_fich   }
{-----}
procedure autr_fich;
begin
  window(1,8,80,25); clrscr; window(1,1,80,25);
  gotoxy(30,8); write('1. Cote du lac');
  gotoxy(30,9); write('2. Remplissage ou non du lac');
  gotoxy(30,10); write('3. Surfaces cultivées');

```

Nom du programme : b:gestfich.pas

Heure : 12:23

```

=====

gotoxy(30,11); write('4. Besoins en eau');
gotoxy(30,12); write('5. Efficience irrigation'); textcolor(yellow);
gotoxy(30,13); write('6. Menu Précédent'); textcolor(lightcyan);
gotoxy(34,21); write('Votre choix ? ');
end;
{-----}
{      c_menu_creat_fich      }
{-----}
procedure c_menu_creat_fich;
begin
  repeat
    ch:=readkey;
  until ch in ['0'..'9'];
  case ch of
    '1':begin
      rmc:='REJETS AU LAC'; un:='Mm3/jour'; end;
    '2':begin
      rmc:='PRECIPITATIONS'; un:='mm'; end;
    '3':begin
      rmc:='APPORTS DU PLEUVE'; un:='Mm3/jour'; end;
    '4':begin
      rmc:='POMPAGES CSS'; un:='Mm3/jour'; end;
    '5':begin
      rmc:='POMPAGES SONEBS'; un:='Mm3/jour'; end;
    '6':begin
      rmc:='POMPAGES CANAL DU CAYOR'; un:='Mm3/jour'; end;
    '7':begin
      rmc:='POMPAGES SUPPLEMENTAIRES'; un:='Mm3/jour'; end;
    '8':begin
      rmc:='EVAPORATION JOURNALIERE'; un:='mm/jour'; end;
    '9':begin
      autr_fich;
      repeat
        ch:=readkey;
      until ch in ['1'..'6'];
      case ch of
        '1':begin
          rmc:='COTE DU LAC'; un:='m IGN'; end;
        '2':begin
          rmc:='REMPLISSAGE OU NON DU LAC'; un:='Sans unités'; end;
        '3':begin
          rmc:='SURFACES CULTIVEES'; un:='ha'; affiche; end;
        '4':begin
          rmc:='BESOINS EN EAU'; un:='m3/ha/jour'; affiche; end;
        '5':begin
          rmc:='EFFICIENCE IRRIGATION'; un:='%'; affiche; end;
        '6':begin
          menu_creat_fich; c_menu_creat_fich; end;
      end;
    end;
  end;
end;
{-----}
{      creat_fich      }
{      crée un fichier typé      }
{-----}
procedure creat_fich;
begin

```

```
=====
clrscr;
for i:=1 to 80 do write('-');
gotoxy(3,2); writeln('CREATION D''UN FICHIER ',rnc);
for i:=1 to 80 do write('-');
gotoxy(10,12); write('Confirmer la création du fichier (O/N)');
ch:=readkey;
if ch in ['o','O']
then begin
    gotoxy(1,12); clrscr; t_curseurVisibl;
    gotoxy(13,7); write('Nom du fichier ..... ');
    gotoxy(46,7); readln(nom);
    gotoxy(13,10); write('Nombre de données à entrer .. ');
    readln(nbre_jours); t_curseurInvis;
    assign(f, nom); rewrite(f);
    window(1,4,80,25); clrscr; window(1,1,80,25);
    gotoxy(15,6); write('Introduisez-vous les données journalièrement');
    gotoxy(25,7); write('ou par tranches ? (J/T)'); gotoxy (35,8);
    ch:=readkey;
    if ch in ['J','j']
    then begin
        window(1,4,80,25); clrscr; window(1,1,80,25);
        gotoxy(69,2); write(un);
        textcolor(31); gotoxy(1,4); write(nom:80); textcolor(11);
        gotoxy(12,8); write('Introduisez vos données journalièrement');
        t_curseurVisibl;
        for i:=1 to nbre_jours do
        begin
            gotoxy(18,11); write('valeur n',i:4,' ..... ');
            gotoxy(44,11); clrscr; readln(element);
            write(f, element);
        end; { for }
    end { then }
else begin
    window(1,4,80,25); clrscr; window(1,1,80,25);
    gotoxy(69,2); writeln(un);
    gotoxy(1,4); textcolor(31); write('PAR TRANCHES');
    write(nom:68); textcolor(11); t_curseurVisibl;
    gotoxy(15,10); write('Nombre de tranches à valeur constante : ');
    readln (ntvc); j:=1;
    window(1,5,80,25); clrscr; window(1,1,80,25); entetel;
    if dt<>1
    then begin
        window(1,5,80,25); clrscr; window(1,1,80,25);
        introduction1;
        window(1,5,80,25); clrscr; window(1,1,80,25);
        introduction2;
    end
    else begin
        window(1,5,80,25); clrscr; window(1,1,80,25);
        introduction2;
    end;
for j:=2 to ntvc do
begin
    window(1,5,80,25); clrscr; window(1,1,80,25); entetel;
    if dt<>(filesize(f)+1)
    then begin
        window(1,5,80,25); clrscr; window(1,1,80,25);
        introduction1;
```

=====

```
        window(1,5,80,25); clrscr; window(1,1,80,25);
        introduction2;
    end
else begin
    window(1,5,80,25); clrscr; window(1,1,80,25);
    introduction2;
    end;
end; { for }
if nbre_jours<>filesize(f)
then begin
    window(1,5,80,25); clrscr; window(1,1,80,25);
    gotoxy(19,7); writeln('Introduisez vos données');
    for i:=(filesize(f)+1) to nbre_jours do
    begin
        gotoxy(18,10); write('valeur n',i:3,' ..... ');
        gotoxy(43,10); clreol; readln(element);
        write(f, element);
    end; { for }
end; { then }
end; { else }
t_curseurInvis;
close(f);
end;
end;
{-----}
{      correct_fich      }
{      corrige les données d'un fichier typé }
{-----}
procedure correct_fich (nom1 :string);
var
    jour, compt :longint;

begin
    window(1,4,80,25); clrscr; window(1,1,80,25);
    gotoxy(15,6); write(filesize(f):3, ' valeur(s) dans le fichier ');
    write(nom1);
    gotoxy(15,13); write('Voulez-vous corriger certaines données ? (O/N) ');
    ch:=readkey;
    if ch in ['O','o']
    then begin
        repeat
            window(1,1,80,25); clrscr; textcolor(15); gotoxy(27,2);
            write('CORRECTION DE VOS DONNEES'); gotoxy(1,5);
            writeln ('Rappel: ', filesize(f):3, ' valeur(s) dans le fichier ', nom1);
            for i:=1 to 80 do write('-');
            window(1,10,80,25); t_curseurVisibl;
            gotoxy(30,1); write('valeur n ');
            gotoxy(30,2); write('-----');
            repeat
                gotoxy(40,1); clreol;
                {$I-}           { inhibe la detect. d'erreurs }
                readln (jour);
                {$I+}           { rétab. la detect. d'erreurs }
                res:=ioresult;
            until (jour<=filesize(f)) and (jour>=0) and (res=0);
            compt:=jour+25;
            if compt>filesize(f) then compt:=filesize(f);
            t_curseurInvis;
```

=====

```
for i:=jour to compt do
begin
  clrscr;
  gotoxy(30,1); write('valeur n ' ');
  gotoxy(30,2); write('-----');
  gotoxy(40,1); clrscr; writeln(i);
  gotoxy(5,5); seek(f,i-1); read(f,element);
  write(' valeur introduite : ', element:8:3);
  write(' corriger ? (O/N) ');
  ch:=readkey;
  if ch in ['O','o']
  then begin
    t_curseurVisibl;
    gotoxy(40,5); clrscr; write(' nouvelle valeur : ');
    readln(element); gotoxy(40,5); clrscr;
    seek(f,i-1); write(f,element);
    t_curseurInvis;
  end; { if }
end; { for }
textcolor(27); gotoxy(22,10); write('Reprendre la correction ? (O/N) ');
ch:=readkey;
until ch in ['A'..'N','P'..'Z','a'..'n','p'..'z'];
end;
close(f);
textcolor(15);
window(1,1,80,25);
end;
{-----}
{ ajout_fich }
{ ajoute des données à un fichier typé }
{-----}
procedure ajout_fich (noml :string);
var
  compteur :integer;

begin
  window(1,4,80,25); clrscr; window(1,1,80,25);
  gotoxy(15,6); write(filesize(f):3, ' valeur(s) dans le fichier ');
  write(noml);
  gotoxy(15,13); write('Voulez-vous ajouter des données ? (O/N) ');
  ch:=readkey;
  if ch in ['O','o']
  then begin
    clrscr; textcolor(15); gotoxy(27,2); write('AJOUT DE DONNEES');
    gotoxy(1,5);
    writeln ('Rappel: ', filesize(f):3, ' valeur(s) dans le fichier ', noml);
    for i:=1 to 80 do write('-');
    t_curseurVisibl;
    gotoxy(18,10); write('Nombre de données');
    gotoxy(18,11); write('à ajouter au fichier ... ');
    gotoxy(43,11); readln(nbr_don);
    gotoxy(18,13); write('Après quel rang ..... ');
    repeat
      gotoxy(43,13); clrscr;
      {$I-} { inhibe la détect. d'erreurs }
      readln (rang);
      {$I+} { rétab. la détect. d'erreurs }
    res:=ioreadln;
  end;
end;
```

=====

```
    until (rang<=filesize(f)) and (rang>=0) and (res=0);
    seek(f,filesize(f));
    compteur:=filesize(f)-rang;
for i:=1 to nbr_don do
begin
    window(1,17,80,25); clrscr; window(1,1,80,25);
    gotoxy(18,18); write('valeur n' , i:3, ' ..... ');
    gotoxy(18,19); write('-----');
    gotoxy(43,18); clreol; readln(element); write(f,element);
end; { for }
t_curseurInvis;
if rang<>filesize(f)-nbr_don { -----> cas où les données ne }
then begin { sont pas ajoutées à }
    for i:=0 to nbr_don-1 do { la fin du fichier }
    begin
        seek(f,filesize(f)-nbr_don+i); read(f,element);
        for j:=1 to compteur do
        begin
            seek(f,filesize(f)-nbr_don-j+i); read(f,element1);
            seek(f,filesize(f)-nbr_don-j+i); write(f,element);
            seek(f,filesize(f)-nbr_don-j+i+1); write(f,element1);
        end; { for j }
    end; { for i }
end; { then }
end;
close(f);
end;
{-----}
{      suppr_fich      }
{      supprime des données d'un fichier typé }
{-----}
procedure suppr_fich (nom1 :string);
begin
    window(1,4,80,25); clrscr; window(1,1,80,25);
    gotoxy(15,6); write(filesize(f):3, ' valeur(s) dans le fichier ');
    write(nom1);
    gotoxy(15,13); write('Voulez-vous supprimer des données ? (O/N) ');
    ch:=readkey;
    if ch in ['O','o']
    then begin
        clrscr; textcolor(15); gotoxy(29,2);
        write('SUPPRESSION DE DONNEES'); gotoxy(1,5);
        writeln ('Rappel: ', filesize(f):3, ' valeur(s) dans le fichier ', nom1);
        for i:=1 to 80 do write('-');
        t_curseurVisibl;
        gotoxy(18,10); write('Nombre de données');
        gotoxy(18,11); write('à supprimer du fichier ... ');
        gotoxy(45,11); readln(nbr_don);
        gotoxy(18,13); write('Rang du premier');
        gotoxy(18,14); write('élément à supprimer ..... ');
        repeat
            gotoxy(45,14); clreol;
            {$I-} { inhibe la détect. d'erreurs }
            readln(rang);
            {$I+} { rétab. la détect. d'erreurs }
            res:=ioresult;
        until (rang<=filesize(f)) and (rang>0) and (res=0);
```

```
=====

t_curseurInvis;
if nbr_don=1
then begin
  for i:=rang-1 to filesize(f)-2 do
  begin
    seek(f,i); read(f,element); read(f,element1);
    seek(f,i); write(f,element1);
    seek(f,i+1); write(f,element);
  end; { for }
  seek(f,filesize(f)-1); truncate(f);
end
else begin
  for i:=0 to filesize(f)-rang-nbr_don do
  begin
    seek(f,rang+nbr_don-1+i); read(f,element);
    seek(f,rang-1+i); write(f,element);
  end; { for }
  seek(f,filesize(f)-nbr_don); truncate(f);
end; { else }
end;
close (f);
end;
{-----}
{      menu_mod_fich      }
{      menu de modification de fichier      }
{-----}
procedure menu_mod_fich;
begin
  clrscr;
  gotoxy(22,2); write('MODIFICATION DES DONNEES D'UN FICHIER');
  gotoxy(22,3); write('-----');
  gotoxy(28,10); write('1. Correction des données');
  gotoxy(28,11); write('2. Ajout de données');
  gotoxy(28,12); write('3. Suppression de données'); textcolor(yellow);
  gotoxy(28,13); write('4. Retour au menu principal'); textcolor(white);
  gotoxy(34,20); write('Votre choix ? ');
end;
{-----}
{      c_menu_mod_fich      }
{-----}
procedure c_menu_mod_fich;
begin
  repeat
  ch:=readkey;
  until ch in ['1'..'4'];
  window(1,4,80,25); clrscr; window(1,1,80,25);
  case ch of
  '1' :begin
    gotoxy(30,6); write('Correction de données');
    gotoxy(30,7); write('-----');
    gotoxy(5,11); write('Nom du fichier ..... ');
    t_curseurVisibl; gotoxy(31,11); readln(nom);
    t_curseurInvis;
    assign(f,nom);
    {$I-}      { inhibe la detect. d'erreurs }
    reset(f);
    {$I+}      { rétab. la detect. d'erreurs }
    if ioread=0
```

```
=====

    then correct_fich(nom)
  else begin
    message; menu_mod_fich; c_menu_mod_fich;
  end;
end;
'2' :begin
  gotoxy(33,6); write('Ajout de données');
  gotoxy(33,7); write('-----');
  gotoxy(5,11); write('Nom du fichier ..... ');
  t_curseurVisibl; gotoxy(31,11); readln(nom);
  t_curseurInvis;
  assign(f,nom);
  {$I-}      { inhibe la détect. d'erreurs }
  reset(f);
  {$I+}      { rétab. la détect. d'erreurs }
  if ioresult=0
  then ajout_fich(nom)
  else begin
    message; menu_mod_fich; c_menu_mod_fich;
  end;
end;
'3' :begin
  gotoxy(30,6); write('Suppression de données');
  gotoxy(30,7); write('-----');
  gotoxy(5,11); write('Nom du fichier ..... ');
  t_curseurVisibl; gotoxy(31,11); readln(nom);
  t_curseurInvis;
  assign(f,nom);
  {$I-}      { inhibe la détect. d'erreurs }
  reset(f);
  {$I+}      { rétab. la détect. d'erreurs }
  if ioresult=0
  then suppr_fich(nom)
  else begin
    message; menu_mod_fich; c_menu_mod_fich;
  end;
end;
end; { case }
end;
{-----}
{   impr_fich   }
{   imprime le contenu d'un fichier   }
{-----}
procedure impr_fich;
var
  ligne :longint;

begin
  clrscr; textcolor(yellow);
  gotoxy(1,6); clreol;
  gotoxy(29,2); write('IMPRESSION D'UN FICHER');
  gotoxy(29,3); write('-----');
  write (^G^G); textcolor(lightgreen);
  t_ecrit_clign(23,7,'Vérifiez que l'imprimante est prête ...');
  delay(3500); gotoxy(1,7); clreol;
  textcolor(yellow); gotoxy(1,11); write('Nom du fichier... ');
  t_curseurVisibl;
  gotoxy(21,11); clreol; textcolor(lightgreen);
```



Programme source du: jeudi, le 25.07.1991  
Nom du programme : b:gestfich.pas  
Heure : 12:40

Page: 12

=====

```
write(lst,i-1:4,'|':3);
for j:=1 to 10 do
begin
  {$I-}           { inhibe la détect. d'erreurs }
  read(f,element);
  {$I+}           { rétab. la détect. d'erreurs }
  if ioresult=0
  then write(lst, element:13:3);
end; { for j }
end;
writeln(lst);
end; { for i }
write(lst,$18);      { fin du mode minuscule EPSON }
end;
end;
end.
```

Nom du programme : b:lgplg.pas

Heure : 13:35

```

-----
{ program : lgplg.pas }
{ }
{ date : 10/07/91 }
{ compiler : turbo pascal 5.0 }
{ }
{ Mis à jour : 24/07/91 }
{ }
{ Auteur : Noël Dacruz BVORA }
{ }
-----
{ Logiciel de Gestion Prévisionnelle }
{ du Lac de Guiers (LGPLG). }
-----

```

program lgplg;

```

uses crt, graph,
    t_io, t_decl, t_fenetre,
    gestfich, lgplgcb, lgplghg;

```

type

```

    indicel =array[1..9] of real;

```

const

```

t1=' LOGICIEL DE GESTION ';
t2='PREVISIONNELLE DU LAC DE GUIERS';
t3='{S E N E G A L}';
t4='conçu et écrit';
t5='par';
t6='Noël Dacruz BVORA';
t7='Copyright (c) Juillet 1991';
t8='_____';
t9=' Veuillez frapper <Enter> ';
t10='PROJET CRE EQUISEM TS 2 0198 F BDB';
t11='ORSTOM - PUL';
accesBGI='c:\LGPLG'; { chemin d'accès des pilotes BGI }

```

var

```

choix :char;
position :longint;
jour_simu, code_jour,
piloteGraphiq, modeGraphiq, codErreur :integer;
apport, perte, cote, cotel, element2,
prod, vr, pl, vp0, l_evap, v_evap, vf,
vlf, v_cay, v_sup, vp1, vp2, v1, vi, v2,
sbv, val_rempl_exist, somme_surf, somme_produit,
surf_moy, cot_i, cot_f, var_vol, h_quai, h_lac, q_taoe :real;
a, b, c, d, e, g, h, l, m, n, p :file of real;

```

```

-----
{ fonction vol_lac }
{ calcule le volume du lac }
-----

```

function vol\_lac (h:real) :real;

```

begin
    vol_lac:=181.69+169.54*h+39.48*h*h+1.137*h*h*h-1.246*h*h*h*h-0.411*h*h*h*h*h;
end;

```

```

-----
{ fonction surf_lac }

```

=====

```
{      calcule la surface du lac      }
{-----}
function surf_lac (h:real) :real;
begin
  if h<-1.5
  then surf_lac:=121.255+85.194*ln(h+2)
  else if (h>=-1.5) and (h<-0.5)
  then surf_lac:=98.399*exp(0.662*ln(h+2))
  else surf_lac:=58.603+164.822*ln(h+2)
end;
{-----}
{ fonction qn1
{ Fonction Qn=f(hq), débit normal Qn dans la
{ Taoué en fonction de la hauteur d'eau au quai
{ de Richard-Toll, hq
{-----}
function qn1 (hq :real) :real;
begin
  if (hq>=60) and (hq<=80)
  then qn1:=0.01*hq+23.4
  else if (hq>80) and (hq<=100)
  then qn1:=0.11*hq+15.4
  else if (hq>100) and (hq<=120)
  then qn1:=0.36*hq-9.6
  else if (hq>120) and (hq<=140)
  then qn1:=0.5*hq-26.4
  else if hq>140
  then qn1:=0.64*hq-46;
end;
{-----}
{ fonction qn2
{ Courbe de tarage à dénivellée normale
{-----}
function qn2 (hq, hl, q :real) :real;
begin
  qn2:=q*exp(0.57*ln(8/((hq-hl)+5)));
end;
{-----}
{ entete_p
{-----}
procedure entete_p;
begin
  t_creePenetre(1,1,80,25,false,1);
  t_ecrit_clign(29,25,t9);
  t_creePenetre(3,3,77,23,true,2);
  t_creePenetre(22,6,58,20,false,0);
  textcolor(lightcyan);
  t_ecrit(3,1,t1);
  t_ecrit(4,2,t2); t_ecrit(12,3,t3);
  t_ecrit(2,4,t8); t_ecrit(12,7,t4);
  t_ecrit(18,8,t5); t_ecrit(11,9,t6);
  t_ecrit(2,12,t8); t_ecrit(2,13,t10);
  t_ecrit(14,14,t11); t_ecrit(7,15,t7);
  t_curseurInvis;      { le curseur est invisible }
  readln;
  window(1,1,80,25);
end;
{-----}
```

Nom du programme : b:lgplg.pas

Heure : 13:39

=====

```

{ menu_simul
{-----}
procedure menu_simul;
begin
  for i:=1 to 80 do write ('-');
  t_ecrit(44,4,'Nom des fichiers');
  t_ecrit_invers(1,6,' FICHIERS APPORTS ');
  gotoxy(5,7); write('Fleuve..... ');
  gotoxy(5,8); write('Rejets CSS au lac..... ');
  gotoxy(5,9); write('Précipitations..... ');
  t_ecrit_invers(1,11,' FICHIERS PERTES ');
  gotoxy(5,12); write('Pompages CSS..... ');
  gotoxy(5,13); write('Pompages SONBES..... ');
  gotoxy(5,14); write('Pompages Canal du CAYOR..... ');
  gotoxy(5,15); write('Pompages Irrigation..... ');
  gotoxy(5,16); write('Pompages Supplémentaires..... ');
  gotoxy(5,17); write('Evaporation journalière..... ');
  t_ecrit_invers(1,19,' AUTRES FICHIERS ');
  gotoxy(5,20); write('Cote du lac..... ');
  gotoxy(5,21); write('Cote au quai de Richard-Toll..... ');
  gotoxy(5,22); write('Volume évaporé quotidiennement..... ');
end;
{-----}
{ nom_fich
{-----}
procedure nom_fich;
begin
  gotoxy(44,j); clrcol; t_curseurVisibl; readln(nom); t_curseurInvis;
end;
{-----}
{ lect_fich_simul
{-----}
procedure lect_fich_simul;
begin
  j:=8;
  repeat
    nom_fich; assign(b,nom);
    {$I-}
    reset(b);
    {$I+}
    res:=ioresult;
  until res=0;
  j:=j+1;
  repeat
    nom_fich; assign(c,nom);
    {$I-}
    reset(c);
    {$I+}
    res:=ioresult;
  until res=0;
  j:=12;
  repeat
    nom_fich; assign(d,nom);
    {$I-}
    reset(d);
    {$I+}
    res:=ioresult;
  until res=0;

```

Nom du programme : b:lgplg.pas

Heure : 13:41

=====

```
j:=j+1;
repeat
  nom_fich; assign(e,nom);
  {$I-}
  reset(e);
  {$I+}
  res:=ioresult;
until res=0;
j:=j+1;
repeat
  nom_fich; assign(f,nom);
  {$I-}
  reset(f);
  {$I+}
  res:=ioresult;
until res=0;
j:=j+1;
repeat
  nom_fich; assign(p,nom);
  {$I-}
  reset(p);
  {$I+}
  res:=ioresult;
until res=0;
j:=j+1;
repeat
  nom_fich; assign(g,nom);
  {$I-}
  reset(g);
  {$I+}
  res:=ioresult;
until res=0;
j:=j+1;
repeat
  nom_fich; assign(h,nom);
  {$I-}
  reset(h);
  {$I+}
  res:=ioresult;
until res=0;
end;
{-----}
{ ferm_fich }
{-----}
procedure ferm_fich;
begin
  close(a); close(b); close(c); close(d); close(e); close(f);
  close(p); close(g); close(h); close(l); close(m); close(n);
end;
{-----}
{ cadre }
{-----}
procedure cadre;
begin
  textcolor(lightgreen);
  t_ecrit(17,7,'r'); t_horizontal(18,7,47,196); t_ecrit(65,7,'r');
  t_vertical(17,8,13,179); t_vertical(65,8,13,179);
  t_ecrit(17,20,'L'); t_horizontal(18,20,47,196); t_ecrit(65,20,'L');
```

Nom du programme : b:lgplg.pas

Heure : 13:42

=====

```

    textcolor(white);
end;
{-----}
{ calcul_hq
{   calcule la hauteur minimale au quai de
{   Richard-Toll
{-----}
procedure calcul_hq;
begin
  q_taque:=vf*10000/864;
  h_lac:=cote*100;
  if h_lac<60
  then h_quai:=60
  else h_quai:=h_lac;
  repeat
  h_quai:=h_quai+0.05;
  until (abs(qn2(h_quai, h_lac, q_taque)-qn1(h_quai))<=0.01) or (h_quai>161);
  h_quai:=h_quai/100;
  write(m,h_quai);
end;
{-----}
{ surfacemoyenne
{   calcule la surface moyenne pour la période
{   de simulation
{-----}
procedure surfacemoyenne;
begin
  reset(1);
  somme_surf:=0;
  somme_produit:=0;
  for i:=1 to filesize(1) do
  begin
    read(1,cote);
    somme_surf:=somme_surf+surf_lac(cote);
    somme_produit:=somme_produit+cote*surf_lac(cote);
  end;
  surf_moy:=surf_lac(somme_produit/somme_surf);
end;
{-----}
{ bilan
{   présente le bilan hydrologique
{-----}
procedure bilan;
var
  tot_app, tot_pert :real;

begin
  clrscr; textcolor(lightgreen);
  t_ecrit(2,3,'P'); t_horizontal(3,3,76,205); t_ecrit(79,3,'I');
  t_vertical(2,4,19,186); t_vertical(41,8,10,179); t_vertical(79,4,19,186);
  t_ecrit(2,5,'I'); t_horizontal(3,5,76,196); t_ecrit(79,5,'I');
  t_horizontal(31,16,6,196); t_horizontal(68,16,6,196);
  t_ecrit(2,23,'M'); t_horizontal(3,23,76,205); t_ecrit(79,23,'M');
  textcolor(white);
  t_ecrit(17,4,'BILAN HYDROLOGIQUE DU LAC DE GUIERS (en m d'eau)');
  t_ecrit_invers(10,8,' APPORTS '); t_ecrit_invers(47,8,' PBTES ');
  t_ecrit(10,10,'Pluie.....'); t_ecrit(47,10,'CSS.....');
  t_ecrit(10,11,'Rejets.....'); t_ecrit(47,11,'SONBES.....');

```

=====

```
t_ecrit(10,12,'Fleuve.....'); t_ecrit(47,12,'Canal du Cayor.....');
t_ecrit(47,13,'Irrigation.....');
t_ecrit(47,14,'Supplement.....');
t_ecrit(47,15,'Evaporation.....');
t_ecrit(10,17,'Total.....'); t_ecrit(47,17,'Total.....');
t_ecrit(5,21,'Variation de volume du lac.....');
tot_app:=0; tot_pert:=0;
for i:=1 to 9 do
begin
  if i in[1..3]
  then tot_app:=tot_app+poste[i]
  else tot_pert:=tot_pert+poste[i];
end;
textcolor(lightcyan);
gotoxy(31,10); write(poste[1]:5:2); gotoxy(68,10); write(poste[4]:5:2);
gotoxy(31,11); write(poste[2]:5:2); gotoxy(68,11); write(poste[5]:5:2);
gotoxy(31,12); write(poste[3]:5:2); gotoxy(68,12); write(poste[6]:5:2);
gotoxy(68,13); write(poste[7]:5:2);
gotoxy(68,14); write(poste[8]:5:2);
gotoxy(68,15); write(poste[9]:5:2);
gotoxy(31,17); write(tot_app:5:2); gotoxy(68,17); write(tot_pert:5:2);
gotoxy(39,21); write(var_vol:5:2);
textcolor(white);
end;
{-----}
{ calcul_simul1 }
{-----}
procedure calcul_simul1;
begin
  for i:=1 to 9 do poste[i]:=0;
  sbv:=surf_lac(4.0);
  seek(h,code_jour-1);
  write(l,cote); cot_i:=cote;
  for i:=1 to jour_simu do
  begin
    gotoxy(55,18); write(i:5);
    read(a,vf); read(b,vr); read(c,pl); read(d,vpl);
    read(e,vp2); read(f,v_cay); read(p,vi); read(g,v_sup);
    if eof(h) then seek(h,0); {-----}
                                { renvoie le pointeur au debut }
                                { du fichier EVAPORATION si la }
                                { fin du fichier est atteinte }
                                {-----}
    read(h,l_evap);
    if vf<>0
    then calcul_hq
    else begin
      h_quai:=0;
      write(m,h_quai);
    end;
    v_evap:=surf_lac(cote)*l_evap*0.001; write(n,v_evap);
    vp0:=0.001*surf_lac(cote)*pl+0.0001*(sbv-surf_lac(cote))*pl;
    poste[1]:=poste[1]+vp0; poste[2]:=poste[2]+vr; poste[3]:=poste[3]+vf;
    poste[4]:=poste[4]+vpl; poste[5]:=poste[5]+vp2; poste[6]:=poste[6]+v_cay;
    poste[7]:=poste[7]+vi; poste[8]:=poste[8]+v_sup; poste[9]:=poste[9]+v_evap;
    v1:=vol_lac(cote);
    v2:=v1+vf+vr+vp0-vpl-vp2-v_cay-vi-v_sup-v_evap;
    cotel:=cote;
```

=====

```
if v2<v1
then while (vol_lac(cotel)-v2)>0.01 do cotel:=cotel-0.001
else while (v2-vol_lac(cotel))>0.01 do cotel:=cotel+0.001;
write(l,cotel);
cote:=cotel;
end;
cot_f:=cote;
surfacemoyenne;
for i:=1 to 9 do poste[i]:=poste[i]/surf_moy;
var_vol:=cot_f-cot_i;
end;
{-----}
{ simul1 }
{ Cote du lac en fin de période de simulation }
{-----}
procedure simul1;
begin
clrscr; textcolor(lightcyan);
gotoxy(1,1); write('CALCULS :'); textcolor(white); gotoxy(21,1);
writeln('COTE DU LAC A LA FIN DE LA PERIODE DE SIMULATION');
menu_simul;
j:=7;
repeat
nom_fich; assign(a,nom);
{$I-}
reset(a);
{$I+}
res:=ioresult;
until res=0;
lect_fich_simul;
j:=20;
nom_fich; assign(l,nom); rewrite(l); j:=j+1;
nom_fich; assign(m,nom); rewrite(m); j:=j+1;
nom_fich; assign(n,nom); rewrite(n);
readln;
window(1,3,80,25); clrscr; window(1,1,80,25);
cadre; t_curseurVisibl;
gotoxy(20,9); write ('Nombre de jours de simulation..... ');
readln (jour_simu); gotoxy(55,9); write(jour_simu:5);
gotoxy(20,11); writeln ('Code du jour de début');
gotoxy(20,12); write ('de période de simulation..... ');
readln (code_jour); gotoxy(55,12); write(code_jour:5);
gotoxy(20,14); write ('Cote initiale du lac en m..... ');
readln (cote); gotoxy(55,14); write(cote:8:2);
t_ecrit_clign(20,18,'Calcul en cours');
t_curseurInvis;
calcul_simul1;
textcolor(lightgreen);
t_ecrit(17,16,'|'); t_horizontal(18,16,47,196); t_ecrit(65,16,'|');
textcolor(white);
gotoxy(20,18); write('Cote finale en m..... ');
textcolor(lightcyan); write(cote:8:2);
ferm_fich;
readln;
bilan;
readln;
end;
{-----}
```

Nom du programme : b:lgplg.pas

Heure : 13:49

=====

```

{ calcul_simul2
{-----}
procedure calcul_simul2;
begin
  for i:=1 to 9 do poste[i]:=0;
  sbv:=surf_lac(4.0);
  k:=0;
  code_jour:=code_jour-1;
  write (1,cote); cot_f:=cote;
  for i:=jour_simu downto 1 do
  begin
    gotoxy(55,18); write(i:5);
    seek(a,filesize(a)-k-1); seek(b,filesize(b)-k-1); seek(c,filesize(c)-k-1);
    seek(d,filesize(d)-k-1); seek(e,filesize(e)-k-1); seek(f,filesize(f)-k-1);
    seek(p,filesize(p)-k-1); seek(g,filesize(g)-k-1);
    k:=k+1;
    if code_jour<0 {-----}
    then code_jour:=filesize(h)-1; { renvoie le pointeur du fichier }
    {   EVAPORATION sur la dernière   }
    {           valeur           }
    {-----}

    seek(h,code_jour); read(h,l_evap);
    read(a,vf); read(b,vr); read(c,pl); read(d,vpl);
    read(e,vp2); read(f,v_cay); read(p,vi); read(g,v_sup);
    if vf<>0
    then calcul_hq
    else begin
      h_quai:=0;
      write(m,h_quai);
      end;
    poste[2]:=poste[2]+vr; poste[3]:=poste[3]+vf; poste[4]:=poste[4]+vpl;
    poste[5]:=poste[5]+vp2; poste[6]:=poste[6]+v_cay; poste[7]:=poste[7]+vi;
    poste[8]:=poste[8]+v_sup;
    v_evap:=surf_lac(cote)*l_evap*0.001;
    vp0:=0.001*surf_lac(cote)*pl+0.0001*(sbv-surf_lac(cote))*pl;
    v1:=vol_lac(cote);
    v2:=v1-vf-vr-vp0+vpl+vp2+v_cay+vi+v_sup+v_evap;
    cotel:=cote;
    if v2>v1
    then while(v2-vol_lac(cotel))>0.01 do cotel:=cotel+0.001
    else while(vol_lac(cotel)-v2)>0.01 do cotel:=cotel-0.001;
    v_evap:=surf_lac(cotel)*l_evap*0.001;
    vp0:=0.001*surf_lac(cotel)*pl+0.0001*(sbv-surf_lac(cotel))*pl;
    v2:=v1-vf-vr-vp0+vpl+vp2+v_cay+vi+v_sup+v_evap;
    cotel:=cote;
    if v2>vol_lac(cotel)
    then while(v2-vol_lac(cotel))>0.01 do cotel:=cotel+0.001
    else while(vol_lac(cotel)-v2)>0.01 do cotel:=cotel-0.001;
    v_evap:=surf_lac(cotel)*l_evap*0.001;
    vp0:=0.001*surf_lac(cotel)*pl+0.0001*(sbv-surf_lac(cotel))*pl;
    seek(1,filesize(1));
    write(1,cotel);
    write(n,v_evap);
    poste[1]:=poste[1]+vp0;
    poste[9]:=poste[9]+v_evap;
    cote:=cotel;
    code_jour:=code_jour-1;
  end;
end;

```

Nom du programme : b:lgplg.pas

Heure : 13:51

=====

```

cot_i:=cote;
textcolor(lightcyan);
t_écrit(20,17,'Arrangement des données des');
t_écrit(20,18,'fichiers du groupe AUTRES FICHIERS      ');
textcolor(white);
for j:=1 to filesize(l)-1 do
begin
  seek(l,filesize(l)-1); read(l,element1);
  for i:=1 to filesize(l)-j do
  begin
    seek(l,filesize(l)-i-1); read(l,element);
    seek(l,filesize(l)-i); write(l,element);
  end; { for i }
  seek(l,filesize(l)-i-1); write(l,element1);
end; { for j }

for j:=1 to filesize(m)-1 do
begin
  seek(m,filesize(m)-1); read(m,element1);
  seek(n,filesize(n)-1); read(n,valeur);
  for i:=1 to filesize(m)-j do
  begin
    seek(m,filesize(m)-i-1); read(m,element);
    seek(m,filesize(m)-i); write(m,element);
    seek(n,filesize(n)-i-1); read(n,element);
    seek(n,filesize(n)-i); write(n,element);
  end; { for i }
  seek(m,filesize(m)-i-1); write(m,element1);
  seek(n,filesize(n)-i-1); write(n,valeur);
end; { for j }
surfacemoyenne;
for i:=1 to 9 do poste[i]:=poste[i]/surf_moy;
var_vol:=cot_f-cot_i;
end;
{-----}
{ simul2 }
{ Cote du lac en début de période de simulation }
{-----}
procedure simul2;
begin
  clrscr; textcolor(lightcyan);
  gotoxy(1,1); write('CALCULS :'); textcolor(white); gotoxy(21,1);
  writeln('COTE DU LAC AU DEBUT DE LA PERIODE DE SIMULATION');
  menu_simul;
  j:=7;
  repeat
    nom_fich; assign(a,nom);
    {$I-}
    reset(a);
    {$I+}
    res:=ioresult;
  until res=0;
  lect_fich_simul;
  j:=20;
  nom_fich; assign(l,nom); rewrite(l); j:=j+1;
  nom_fich; assign(m,nom); rewrite(m); j:=j+1;
  nom_fich; assign(n,nom); rewrite(n);
  readln;

```

Nom du programme : b:lgplg.pas

Heure : 13:53

```

=====

window(1,3,80,25); clrscr; window(1,1,80,25);
cadre; t_curseurVisibl;
gotoxy(20,9); write ('Nombre de jours de simulation..... ');
readln (jour_simu); gotoxy(55,9); write(jour_simu:5);
gotoxy(20,11); writeln ('Code du jour de fin');
gotoxy(20,12); write ('de période de simulation..... ');
readln (code_jour); gotoxy(55,12); write(code_jour:5);
gotoxy(20,14); write ('Cote finale du lac en m..... ');
readln (cote); gotoxy(55,14); write(cote:8:2);
t_ecrit_clign(20,18,'Calcul en cours');
t_curseurInvis;
calcul_simul2;
textcolor(lightgreen);
t_ecrit(17,16,'|'); t_horizontal(18,16,47,196); t_ecrit(65,16,'|');
gotoxy(20,17); clreol; t_ecrit(65,17,'|');
textcolor(white);
gotoxy(20,18); write('Cote initiale en m..... ');
textcolor(lightcyan); write(cot_i:8:2);
ferm_fich;
readln;
bilan;
readln;
end;
{-----}
{ calcul_simul3 }
{-----}
procedure calcul_simul3;
begin
sbv:=surf_lac(4.0);
seek(h,code_jour-1);
read(m,element); write(l,element);
cote:=element;
for i:=1 to jour_simu do
begin
gotoxy(55,18); write(i:5);
read(b,vr); read(c,pl); read(d,vp1); read(e,vp2);
read(f,v_cay); read(p,vi); read(g,v_sup); read(n,val_rempl_exist);
if eof(h) then seek(h,0); {-----}
{ renvoie le pointeur au début }
{ du fichier EVAPORATION si la }
{ fin du fichier est atteinte }
{-----}

read(h,l_evap);
v_evap:=surf_lac(cote)*l_evap*0.001;
vp0:=0.001*surf_lac(cote)*pl+0.0001*(sbv-surf_lac(cote))*pl;
v1:=vol_lac(cote);
read(m,element1);
if val_rempl_exist=0
then begin
v2:=v1+vr+vp0-vp1-vp2-v_cay-vi-v_sup-v_evap;
cotel:=cote;
if v2<v1
then while(vol_lac(cotel)-v2)>0.01 do cotel:=cotel-0.001
else while(v2-vol_lac(cotel))>0.01 do cotel:=cotel+0.001;
write(l,cotel);
cote:=cotel;
if v2<vol_lac(element1)
then begin { calcul d'un apport complémentaire }

```

=====

```

v_evap:=surf_lac(element)*l_evap*0.001;
vp0:=0.001*surf_lac(element)*pl+0.0001*(sbv-surf_lac(element))*pl;
if vol_lac(element1)>=vol_lac(element)
then vf:=vol_lac(element1)-vol_lac(element)-vr-vp0+vp1+vp2+v_cay+vi+v_sup+v_evap
[vlf] else vf:=-[vol_lac(element)-vol_lac(element1)+vp0+vr-vp1-vp2-v_cay-vi-v_sup-v_evap];
write(a,vf);
end
else if (v2>vol_lac(element1)) and (vol_lac(element1)<=vol_lac(element))
then begin { on pourrait fournir de l'eau à Diana }
v_evap:=surf_lac(element)*l_evap*0.001;
vp0:=0.001*surf_lac(element)*pl+0.0001*(sbv-surf_lac(element))*pl;
vlf:=vol_lac(element1)-vol_lac(element)-vr-vp0+vp1+vp2+v_cay+vi+v_sup+v_evap;
write(a,vlf);
end
else if (v2>vol_lac(element1)) and (vol_lac(element1)>vol_lac(element))
then begin
v_evap:=surf_lac(element)*l_evap*0.001;
vp0:=0.001*surf_lac(element)*pl+0.0001*(sbv-surf_lac(element))*pl;
vf:=vol_lac(element1)-vol_lac(element)-vr-vp0+vp1+vp2+v_cay+vi+v_sup+v_evap;
write(a,vf);
end
else if v2=vol_lac(element1) { les cotes sont identiques }
then begin
vf:=0;
write(a,vf);
end
end
else begin { if val_rempl_exist=1 }
v_evap:=surf_lac(element)*l_evap*0.001;
vp0:=0.001*surf_lac(element)*pl+0.0001*(sbv-surf_lac(element))*pl;
vf:=vol_lac(element1)-vol_lac(element)-vp0-vr+vp1+vp2+v_cay+vi+v_sup+v_evap;
write(a,vf);
v2:=v1+vr+vp0+vf-vp1-vp2-v_cay-vi-v_sup-v_evap;
cotel:=cote;
while(v2-vol_lac(cotel))>0.01 do cotel:=cotel+0.001;
write(l,cotel);
cote:=cotel;
end;
element:=element1;
end; { for i }
end;
{-----}
{ simul3 }
{ Réajustement de la cote du lac }
{-----}
procedure simul3;
begin
clrscr; textcolor(lightcyan);
gotoxy(1,1); write('CALCULS '); textcolor(white); gotoxy(20,1);
writeln('REAJUSTEMENT DU TYPE D'EVOLUTION DE LA COTE DU LAC');
menu_simul;
gotoxy(5,20); write('Cotes du lac réajustées..... ');
gotoxy(5,21); write('Evolution choisie de la cote du lac... ');
gotoxy(5,22); write('Remplissage ou non du lac..... ');
j:=7;
nom_fich; assign(a,nom); rewrite(a);
lect_fich_simul;
j:=20;

```

Nom du programme : b:lgplg.pas

Heure : 13:58

=====

```

nom_fich; assign(l,nom); rewrite(l); j:=j+1;
repeat
  nom_fich; assign(m,nom);
  {$I-}
  reset(m);
  {$I+}
  res:=ioresult;
until res=0;
j:=j+1;
repeat
  nom_fich; assign(n,nom);
  {$I-}
  reset(n);
  {$I+}
  res:=ioresult;
until res=0;
readln;
window(1,4,80,25); clrscr; window(1,1,80,25);
cadre; t_curseurVisibl;
gotoxy(20,9); write('Nombre de jours de simulation..... ');
readln(jour_simu); gotoxy(55,9); write(jour_simu:5);
gotoxy(20,11); writeln('Code du jour de début');
gotoxy(20,12); write('de période de simulation..... ');
readln(code_jour); gotoxy(55,12); write(code_jour:5);
t_ecrit_clign(20,18,'Calcul en cours');
t_curseurInvis;
calcul_simul3;
textcolor(lightcyan);
gotoxy(20,18); write('fin du calcul'); clreol;
textcolor(lightgreen); t_ecrit(65,18,'|');
ferm_fich;
readln;
end;
{-----}
{ calcul_simul4 }
{-----}
procedure calcul_simul4;
begin
  apport:=0; perte:=0;
  sbv:=surf_lac(4.0);
  seek (h,code_jour-1);
  for i:=1 to jour_simu do
  begin
    gotoxy(55,18); write(i:5);
    read(m,val_rempl_exist);
    read(b,vr); read(c,pl); read(d,vp1);
    read(e,vp2); read(f,v_cay); read(p,vi); read(g,v_sup);
    read(l,cote);
    if eof(h) then seek(h,0); {-----}
                                { renvoie le pointeur au début }
                                { du fichier EVAPORATION si la }
                                { fin du fichier est atteinte }
                                {-----}

    read(h,l_evap);
    v_evap:=surf_lac(cote)*l_evap*0.001;
    write(n,v_evap);
    if val_rempl_exist=0
    then begin

```

Nom du programme : b:lgplg.pas

Heure : 14:00

=====

```

        vf:=0; write(a,vf);
    end
else begin
    vp0:=0.001*surf_lac(cote)*pl+0.0001*(sbv-surf_lac(cote))*pl;
    v1:=vol_lac(cote);
    read(1,cote);
    v2:=vol_lac(cote);
    vf:=v2-v1-vp0-vr+vp1+vp2+v_cay+vi+v_sup+v_evap;
    if vf<0
    then begin
        vlf:=v1-v2+vp0+vr-vp1-vp2-v_cay-vi-v_sup-v_evap;
        perte:=perte+vlf; vlf:=-vlf;
        write(a,vlf);
    end
    else begin
        apport:=apport+vf;
        write(a,vf);
    end;
    position:=filepos(1);
    seek(1,position-1);
end;
end; { for i }
end;
{-----}
{ simul4 }
{ Calcul de l'apport fluvial et des pertes en }
{ eau vers la réserve de Diana }
{-----}
procedure simul4;
begin
    clrscr; textcolor(lightcyan);
    gotoxy(1,1); write('CALCULS :'); textcolor(white); gotoxy(27,1);
    writeln('APPORT FLUVIAL ET TRANSPERT VERS DIAMA');
    menu_simul;
    gotoxy(5,21); write('Remplissage ou non du lac..... ');
    j:=7;
    nom_fich; assign(a,nom); rewrite(a);
    lect_fich_simul;
    j:=20;
    repeat
        nom_fich; assign(l,nom);
        {$I-}
        reset(l);
        {$I+}
        res:=ioresult;
    until res=0;
    j:=j+1;
    repeat
        nom_fich; assign(m,nom);
        {$I-}
        reset(m);
        {$I+}
        res:=ioresult;
    until res=0;
    j:=j+1;
    nom_fich; assign(n,nom); rewrite(n);
    readln;
    window(1,3,80,25); clrscr; window(1,1,80,25);

```

=====

```
cadre; t_curseurVisibl;
gotoxy(20,9); write('Nombre de jours de simulation..... ');
readln(jour_simu); gotoxy(55,9); write(jour_simu:5);
gotoxy(20,11); writeln('Code du jour de début');
gotoxy(20,12); write('de période de simulation..... ');
readln(code_jour); gotoxy(55,12); write(code_jour:5);
t_ecrit_clign(20,18,'Calcul en cours');
t_curseurInvis;
calcul_simul4;
textcolor(lightgreen);
t_ecrit(17,14,'|'); t_horizontal(18,14,47,196); t_ecrit(65,14,'|');
textcolor(white);
gotoxy(20,16); write('Apport fluvial en Mm3..... ');
textcolor(lightcyan); write(apport:8:2); textcolor(white);
gotoxy(20,18); write('Transfert vers Diana en Mm3..... ');
textcolor(lightcyan); write(perte:8:2);
ferm_fich;
readln;
end;
{-----}
{ bes_menu }
{-----}
procedure bes_menu;
begin
gotoxy(1,7); write(i:2,'. ',rnc);
gotoxy(7,10); write('1. Fichier surfaces cultivées..... ');
gotoxy(7,11); write('2. Fichier besoins en eau..... ');
gotoxy(7,12); write('3. Fichier efficience irrigation.... ');
end;
{-----}
{ produit }
{-----}
procedure produit;
begin
if i=1
then prod:=((element*element1)/(element2/100))*0.000001
else prod:=prod+(((element*element1)/(element2/100))*0.000001);
end;
{-----}
{ ent_birrig }
{-----}
procedure ent_birrig;
begin
clrscr; textcolor(lightcyan);
gotoxy(1,1); write('CALCULS :'); textcolor(white); gotoxy(29,1);
writeln('BESOINS EN EAU POUR L''IRRIGATION');
for i:=1 to 80 do write ('-');
end;
{-----}
{ item }
{-----}
procedure item;
begin
j:=10;
repeat
nom_fich; assign(a,nom);
{$I-}
reset(a);
```

=====

```
{ $I+ }
res:=ioresult;
until res=0;
j:=j+1;
repeat
nom_fich; assign(b,nom);
{ $I- }
reset(b);
{ $I+ }
res:=ioresult;
until res=0;
j:=j+1;
repeat
nom_fich; assign(c,nom);
{ $I- }
reset(c);
{ $I+ }
res:=ioresult;
until res=0;
for k:=1 to jour_simu do
begin
read(a,element); read(b,element1); read(c,element2);
if i<>1
then read(d,prod);
produit;
if i=1
then write(d,prod)
else begin
position:=filepos(d)-1; seek(d,position); write(d,prod); end;
end;
close(a); close(b); close(c); reset(d);
end;
{-----}
{ canne }
{-----}
procedure canne;
begin
rnc:='CANNE A SUCRE';
t_curseurVisibl;
gotoxy(10,5); write('Nombre de jours..... ');
gotoxy(2,8); write('BESOINS TOTAUX DE L'IRRIGATION');
gotoxy(10,10); write('Fichier..... ');
gotoxy(44,5); readln(jour_simu);
j:=10;
nom_fich; assign(d,nom); rewrite(d);
window(1,10,80,25);
gotoxy(2,4); write('LECTURE DES FICHIBRS');
bes_menu;
item;
readln; window(1,3,80,25); clrscr; window(1,1,80,25);
end;
{-----}
{ riz_h }
{-----}
procedure riz_h;
begin
rnc:='RIZ HIVERNAGE';
gotoxy(2,4); write('LECTURE DES FICHIBRS');
```

Nom du programme : b:lgplg.pas

Heure : 14:06

=====

```
bes_menu;
item;
window(1,10,80,25);
end;
{-----}
{ riz_cs }
{-----}
procedure riz_cs;
begin
  rnc:='RIZ CONTRE-SAISON';
  bes_menu;
  item;
  readln; window(1,3,80,25); clrscr; window(1,1,80,25);
end;
{-----}
{ mais_h }
{-----}
procedure mais_h;
begin
  rnc:='MAIS HIVERNAGE';
  gotoxy(2,4); write('LECTURE DES FICHIBRS');
  bes_menu;
  item;
  window(1,10,80,25);
end;
{-----}
{ mais_cs }
{-----}
procedure mais_cs;
begin
  rnc:='MAIS CONTRE-SAISON';
  bes_menu;
  item;
  readln; window(1,3,80,25); clrscr; window(1,1,80,25);
end;
{-----}
{ legume }
{-----}
procedure legume;
begin
  rnc:='LEGUMBS';
  gotoxy(2,4); write('LECTURE DES FICHIBRS');
  bes_menu;
  item;
  window(1,10,80,25);
end;
{-----}
{ tomate }
{-----}
procedure tomate;
begin
  rnc:='TOMATES';
  bes_menu;
  item;
  readln; window(1,3,80,25); clrscr; window(1,1,80,25);
end;
{-----}
{ divers }
```

```
=====

{-----}
procedure divers;
begin
  rmc:='DIVERS';
  gotoxy(2,4); write('LECTURE DES FICHIERS');
  bes_menu;
  item;
end;
{-----}
{ calcul_birrig }
{ calcul des besoins en eau pour l'irrigation }
{-----}
procedure calcul_birrig;
begin
  for i:=1 to 8 do
  begin
    case i of
      1 :canne;
      2 :riz_h;
      3 :riz_cs;
      4 :mais_h;
      5 :mais_cs;
      6 :legume;
      7 :tomate;
      8 :divers;
    end;
  end;
end;
{-----}
{ birrig }
{-----}
procedure birrig;
begin
  ent_birrig;
  calcul_birrig;
  t_curseurInvis;
  close(d);
  t_ecrit(7,20,'Fin du calcul');
  readln;
end;
{-----}
{ menu_graph }
{-----}
procedure menu_graph;
begin
  t_config_ecran(1,15,1,true);
  t_curseurInvis;
  gotoxy(31,2); write('G R A P H I Q U E S');
  gotoxy(31,3); write('-----');
  gotoxy(27,7); write('1. Courbe');
  gotoxy(27,9); write('2. Histogramme'); textcolor(yellow);
  gotoxy(27,11); write('3. Retour au menu principal'); textcolor(white);
  gotoxy(34,14); write('Votre choix ? ');
end;
{-----}
{ menu_graph_cb }
{-----}
procedure menu_graph_cb;
```

=====

```
begin
  t_curseurVisibl;
  gotoxy(3,10); write ('Fichier..... '); readln(nom);
  gotoxy(3,13); write ('Titre du graphique... '); readln(titre);
  gotoxy(3,16); write ('Titre de l'axe y..... '); readln(titre_y);
  t_curseurInvis;
end;
{-----}
{ menu_graph_bg }
{-----}
procedure menu_graph_bg;
begin
  t_curseurVisibl;
  gotoxy(3,10); write ('Titre du graphique... '); readln(titre);
  t_curseurInvis;
end;
{-----}
{ instal_m_graphiq }
{ Installation du mode graphique et }
{ reconnaissance automatique du pilote }
{-----}
procedure instal_m_graphiq;
begin
  piloteGraphiq:=0;
  InitGraph(piloteGraphiq,modeGraphiq,accesBGI);
  codErreur:=GraphResult;
end;
{-----}
{ c_menu_graph }
{-----}
procedure c_menu_graph;
begin
  repeat
    ch:=readkey;
  until ch in ['1'..'3'];
  case ch of
    '1':begin
      window(1,4,80,25); clrscr; window(1,1,80,25);
      menu_graph_cb;
      assign(f,nom);
      {$I-}
      reset(f);
      {$I+}
      if ioreult=0
      then begin
        instal_m_graphiq;
        if codErreur <> 0
        then begin
          clrscr;
          gotoxy(20,11); write('Le pilote n'est pas installé ...');
          gotoxy(20,13);
          write('Code de l'erreur ..... ',codErreur);
          gotoxy(20,15); writeln(grapherrormsg(codErreur));
          writeln; writeln; writeln; readln;
        end { then }
      else courbe(titre,titre_y);
      end
    else begin
```

=====

```
        message;
        menu_graph;
        c_menu_graph;
    end;
end;
'2':begin
    window(1,4,80,25); clrscr; window(1,1,80,25);
    menu_graph_hg;
    max_n;
    instal_m_graphiq;
    if codErreur (<> 0
    then begin
        clrscr;
        gotoxy(20,11); write('Le pilote n'est pas installé ...');
        gotoxy(20,13);
        write('Code de l''erreur ..... ',codErreur);
        gotoxy(20,15); writeln(grapherrormsg(codErreur));
        writeln; writeln; writeln; readln;
    end { then }
    else begin
        constant;
        fond(titre);
        stylaxe;
        axe_x_h;
        axe_y_h;
        legende;
        histogramme;
    end;
end;
end;
end;
{-----}
{ menu_calc
}
{-----}
procedure menu_calc;
begin
    t_config_ecran(1,15,0,true); gotoxy(15,4); write('CALCULS');
    gotoxy(15,5); write('=====');
    textcolor(lightcyan);
    gotoxy(15,8); write('1. Cote du lac à la fin de la période de simulation');
    gotoxy(15,10); write('2. Cote du lac au début de la période de simulation');
    gotoxy(15,12); write('3. Réajustement du type d''évolution de la cote du lac');
    gotoxy(15,14); write('4. Apport fluvial et transfert vers DIANA');
    gotoxy(15,16); write('5. Besoins en eau pour l''irrigation');
    textcolor(yellow);
    gotoxy(15,18); write('6. Retour au menu principal');
    textcolor(white);
    gotoxy(22,22); write('Votre choix ? ');
end;
{-----}
{ c_menu_calc
}
{-----}
procedure c_menu_calc;
begin
    repeat
        ch:=readkey;
    until ch in ['1'..'6'];
    case ch of
```

=====

```
'1' :simul1;
'2' :simul2;
'3' :simul3;
'4' :simul4;
'5' :birrig;
end; { case }
end;
{-----}
{ menu
{ affiche le menu principal
{-----}
procedure menu;
begin
t_config_ecran(1,10,1,true); textcolor(lightgreen);
gotoxy(34,4); write('MENU PRINCIPAL');
gotoxy(34,5); write('=====');
gotoxy(24,8); write('(1) Création d'un fichier de données');
gotoxy(24,10); write('(2) Modification des données d'un fichier');
gotoxy(24,12); write('(3) Impression d'un fichier');
gotoxy(24,14); write('(4) Calculs');
gotoxy(24,16); write('(5) Graphiques'); textcolor(yellow);
gotoxy(24,18); write('(6) Q U I T T E R'); textcolor(lightgreen);
gotoxy(34,21); write('Votre choix ? ');
end;
{-----}
{ c_menu
{-----}
procedure c_menu;
begin
repeat
choix:=readkey;
until choix in ['1'..'6'];
case choix of
'1' :begin
menu_creat_fich;
c_menu_creat_fich;
if ch='0'
then begin
menu;
c_menu;
end
else begin
repeat
creat_fich;
menu_creat_fich;
c_menu_creat_fich;
until ch='0';
menu;
c_menu;
end;
end;
'2' :begin
clrscr;
t_ecrit_clign(34,9,'Attention !'); textcolor(white);
gotoxy(24,12); write('Le fichier à modifier doit avoir été');
gotoxy(24,13); write('créé auparavant par le programme');
gotoxy(24,16); write('Confirmer l'option choisie (O/N)');
ch:=readkey;
```

```
=====
if ch in ['0','o']
then begin
  repeat
    menu_mod_fich;
    c_menu_mod_fich;
  until ch='4';
  menu;
  c_menu;
end
else begin
  menu;
  c_menu;
end;
end;
'3' :begin
  repeat
    impr_fich;
    clrscr; textcolor(yellow);
    gotoxy(23,6); write('Un autre fichier à imprimer (O/N) ? ');
    ch:=readkey
  until ch in ['N','n'];
  menu;
  c_menu;
end;
'4' :begin
  repeat
    menu_calc;
    c_menu_calc;
  until ch='6';
  menu;
  c_menu;
end;
'5' :begin
  repeat
    menu_graph;
    c_menu_graph;
  until ch='3';
  menu;
  c_menu;
end;
end; { case }
end;
{-----}
{ programme principal }
{-----}
begin
entete_p;
menu;
c_menu;
t_config_ecran(1,10,1,true); textcolor(lightgreen);
gotoxy(23,12); write (' R E T O U R A U D O S ');
readln;
t_config_ecran(1,15,0,true);
end.
```