

Correction du Devoir de Contrôle n°1

Ex 1 :

Q1 :

- ❖ Writeln(F,C1,C2,M[i]) ; Writeln(C1.Sat) ; pointer(F,i-1) ; Readln(M[i]) ; C1.Sat := M[2].Astra; n := filepos(F); ➔ **TRUE**
- ❖ Append(F); ➔ **FALSE** : cette procédure est utilisée avec les fichiers **texte** seulement.
- ❖ Trunc(F) ; ➔ **FALSE** : on utilise Trunc avec les réels et Truncate avec les fichiers typés.
- ❖ Rename(F,F1) ; **FALSE** : on doit renommer le nom physique du fichier et non le nom logique.

Q2 :

Utilisation des directives de compilation {\$I-}{\$I+} lors de l'ouverture du fichier F, pour éviter le plantage du programme, suite à une lecture d'une donnée inexistante.
 ➔ assign(F,'C:\Devoir de controle1\fcanal.dat') ;
 {\$I-}Reset(F) ; {\$I+}
 If (IOresult <> 0) then
 Writeln('Erreur de LECTURE');
 Else
 Begin
 While not(EOF(F)) do

Read(F,C);
 Close(F);
 End;
Q3 :
 Rewrite (F) ;
 For i := 1 to n do
 Begin
 If (M[i].Sat = Astra) then
 Write(F, M[i])
 end;

Ex 2 :

1) Algorithme de la fonction rechercher :

```
0) DEF FN rechercher (var f: fiche , Cr : Canal) :
entier
1) ouvrir(f)
   Res ← -1
   Tant Que (Non(Fin_Fichier(f))) ET (res=-1)
   Faire
       Lire(f,C)
       Si (C.nom=Cr.nom) alors
           Res ← position_fichier(f)-1
       Fin si
   Fin TantQue
```

```
2) Rechercher ← res
3) Fin rechercher
```

2) Algorithme de la procédure suppression :

```
0) DEF Proc Supprimer (var f :fiche , Cs : Canal)
1) p ← rechercher(f,Cs)
2) Si (p ≠ -1) alors
   ouvrir(f)
   creation(ft)
   Tant Que Non(Fin_Fichier(f)) Faire
   Lire(f,C)
   Si (C.nom <> Cs.nom) alors
```

```
Ecrire(ft,C) Fin Si
   Fin Tant Que
   Recréer(f)
   ouvrir(ft)
   Tant Que Non (Fin_fichier(ft)) Faire
   Lire(ft,C)
   Ecrire(f,C)
   Fin Tant Que
   Sinon
   Ecrire ("Donnée inexistante") Fin si
3) Fin Supprimer
```

T.D.O :

Objet	Type / Nature	Rôle
res	Entier	Position (d'un canal) retournée par la fonction rechercher (res = -1 si le Canal n'existe pas dans le fichier F)
C	Canal	Enregistrement Canal représentant une chaîne télévisée du fichier F.
P	Entier	Position (d'un canal) retournée par la fonction rechercher (res = -1 si le Canal n'existe pas dans le fichier F)
ft	fcanal	Fichier de Canal (fichier temporaire)

Ex 3 :

1) Analyse et algorithme du programme principal :

➤ Analyse du P.P :

Résultat : Afficher(p)

Traitement :

```
Proc remplirF2(f1,f2)
[f2= ] associer(f2,"C:\ expire.txt")
Proc remplirF1(f1,p)
[f1 = ]associer(f1,"C:\ articles.dat")
[p=] Proc saisir(p)
```

Fin Analyse.

Algorithme du P.P :

0) Début ex3

- 1) Saisir(p)
- 2) Associer("C:\ articles.dat")
- 3) RemplirF1(f1,p)
- 4) Associer(f2,"C:\ expire.txt")
- 5) RemplirF2(f1,f2)
- 6) Afficher(p)
- 7) Fin

➤ T.D.N.T :

Type
Date = Enregistrement
Jour : 1..31
Mois : 1..12
Annee : 2013..2015
Fin date
Produit = enregistrement
Cod : entier
Nom : chaine de caractères
Prix : réel
Dfab, dexp : Date
Fin Produit
Fiche = fichier de Produits

➤ T.D.O :

Objet	Type / Nature	Rôle
P	Produit	Enregistrement produit (informations d'un article)
F1	Fiche	Fichier typé contenant les produits du magasin
F2	Texte	Fichier texte contenant le nbr de jr avant l'expiration d'un article
Saisir	Procédure	Saisir les informations d'un article
remplirF1	Procédure	Ecriture des articles dans F1
remplirF2	Procédure	Remplissage du fichier F2
Afficher	Procédure	Affichage des produits qui expirent cette semaine

2) Analyses des modules :

➤ Analyse de la procédure saisir:

DEF Proc Saisir (var p : produit)

Résultat : saisir un produit p

Traitement : Ecrire("donner les les infos d'un produit :")

Avec p faire

Cod : donner le code

Répéter

Prix : donner le prix

Jusqu'à (prix > 0)

Nom : donner le nom

Dfab : donner la date de fabrication (jour, mois, année)

Dexp : donner la date d'expiration (jour, mois, année)

Fin Avec

Fin Analyse.

T.D.O

Objet	Type / Nature	Role

➤ **Analyse de la procédure remplirF1:**

DEF Proc remplirF1(var F1 : fiche, p : produit)

Résultat : f1 rempli

Traitement : [recréer (f1)]

S ← 0

Tant que (S <= Sp) faire

Saisir(p)

S ← S + p.prix

Ecrire (f1,p)

Fin tantque

Fermer(f1)

Fin Analyse.

➤ **Analyse de la procédure remplirF2:**

DEF Proc remplirF2(var F1 : fiche, f2 : texte)

Résultat : f2 rempli

Traitement : [ouvrir(f1), pointer(f1,0), recréer(f2)]

Tant que NON (fin_fichier(f1)) faire

Lire (f1,p)

Avec p faire

Nbj ← dfab – dnow

Ecrire_nl(f2, code," ", nom , " reste ", nbj , " pour expirer ")

Fin avec

Fin tantque

Dnow : donner la date d'aujourd'hui

Fermer(f1)

Fermer(f2)

Fin Analyse.

➤ **Analyse de la procédure Afficher:**

DEF Proc afficher (p : produit)

Résultat : afficher le code et le nom des produits expirant cette semaine

Traitement : [ouvrir(f1)]

Tant que Non(Fin_fichier(f1)) faire

Lire (f1,p)

Avec p faire

Si (dexp – dnow) < 7 Alors

Ecrire (cod , " ", nom)

Fin avec

Fin tantque

Fermer(f1)

Fin Analyse.

T.D.O

Objet	Type / Nature	Rôle
S	Réel	Somme des prix des articles
Sp	Constante = 3000	Somme fixée d'avance

Le calcul du nombre des jours avant l'expiration d'un article se fait selon la formule suivante :

$$\text{nbj} \leftarrow (\text{dexp.annee} - \text{dnow.annee}) * 365 + (\text{dexp.mois} - \text{dnow.mois}) * 31 + (\text{dexp.jour} - \text{dnow.jour})$$

T.D.O

Objet	Type / Nature	Rôle
Nbj	Entier	Nombre de jours
Dnow	Date	Date d'aujourd'hui

T.D.O

Objet	Type / Nature	Rôle
Dnow	Date	Date d'aujourd'hui