

Correction du Devoir de Contrôle n°1 (les absents)

Ex 1 :

Q1 : le fichier F est de type Texte → (utilisation de ‘readln(F,y)’).

Q2 : Exécution manuelle de la procédure ‘affiche’

x	readln(f,y)	EOF(f) and (pos(x,y) = 0)	Affiche(x,f)
SI	‘Baccalauriat 2014	False ----- pos = 0	****
	‘Informatique SI ’	False ----- pos = 14	14
	‘Algo, TIC, BD’	False ----- pos = 0	****
	‘Math, Phy SI que’	False ----- pos = 10 ou 0	10

x	readln(f,y)	EOF(f) and (pos(x,y) = 0)	Affiche(x,f)
CA	‘Baccalauriat 2014	False ----- pos = 0 ou 4	4
	‘Informatique SI’	False ----- pos = 0	****
	‘Algo, TIC, BD’	False ----- pos = 0	****
	‘Math, Physique’	False ----- pos = 0	****

Q3 : cette procédure permet d’afficher la position d’une chaîne x si elle existe dans F.

Ex 2 :

V	Le champ d'une variable enregistrement peut lui-même contenir d'autres champs.	
F	On peut afficher le contenu d'une variable enregistrement sans faire référence à ses champs.	il faut faire référence aux champs d'un enregistrement lors de l'affichage.
V	On peut créer des tableaux de variables de type enregistrement.	
F	La syntaxe ReadLn et WriteLn s'applique à tous les types de fichiers.	Readln et Writeln sont utilisées uniquement avec les fichiers texte
V	La procédure Seek permet de positionner le pointeur sur un endroit donné du fichier.	
F	La procédure Reset remet à zéro le contenu d'un fichier.	Reset remet à zéro le pointeur d'un fichier.
F	EOF est une procédure qui indique si la fin d'un fichier est atteinte ou non.	EOF est une fonction qui indique si la fin d'un fichier est atteinte ou non.
F	Append permet d'ajouter un enregistrement à une position donnée d'un fichier texte	Append permet d'ajouter un enregistrement à la fin d'un fichier texte
V	Filesize n'est pas appliquée au fichier texte	
F	La taille d'un fichier est définie d'avance, comme pour le type tableau.	La taille d'un fichier est déterminée après l'écriture des données.

Ex 3 :

1) Analyse et algorithme du programme principal :

➤ **Analyse du P.P :**

Résultat : Transferer(n,T,F)

Traitement :

Proc afficher(n,T)

Proc remplir(n,T,C)

[f =]associer(f,'C:\ bacsport.dat')

Fin Analyse.

➤ **Algorithme du P.P :**

0) **Début ex3**

1) Associer(f,'C:\ bacsport.dat')

2) Remplir(n,T,C)

3) Afficher(n,T)

4) Transferer(n,T,F)

5) Fin

➤ **T.D.N.T :**

Type
TypEx = (course, L.poids, gymnastique)
TypeSport = enregistrement
Exercice = TypEx
Note : réel
Fin TypeSport
Candidat = enregistrement
matricule : entier
Nom & prénom : chaîne
Moy : réel
Sport : TypeSport
Fin Candidat
TAB = tableau de N Candidats
Fiche = fichier de Candidats

Remarque :

Pour l'exercice Gymnastique il faut le définir comme un nouveau type énuméré contenant les valeurs A,B,C et D ➔ un changement de la structure TypEx et TypeSport est donc obligatoire. (voir remarque page 3)

➤ **T.D.O :**

Objet	Type / Nature	Rôle
n	Entier	Nombre de candidats à saisir (dans la procédure remplir)
C	Candidat	Enregistrement regroupant les informations d'un Candidat.
F	Fiche	Fichier typé contenant les Candidats du bac sport
Remplir	Procédure	Remplir le vecteur d'enregistrements T
Afficher	Procédure	Afficher le nombre des candidats qui ont eu des notes > 15 en course
Transférer	Procédure	Transférer les candidats du vecteur T vers le fichier F.

2) Analyses des modules :➤ **Analyse de la****procédure saisir:**

DEF Proc Saisir (var c : candidat)

Résultat : C

Traitement :

Ecrire("donner les info. :")

Avec C faire

Matricule : donner

Nom : donner

Prénom : donner

Sport : donner Exerc

Répéter

note : donner

Jusqu'à (note dans [0..20])

Moy ← formule de calcul

Fin Analyse.

➤ **Analyse de la****procédure remplir:**

DEF Proc remplir(var n : entier, var T: tab)

Résultat : T rempli

Traitement : [n =] répéter

N : donné

Jusqu'à (n > 10 ET n < 30)

Saisir(C), T[1] ← C

Pour i de 2 à n faire

Répéter

Saisir(C)

Jusqu'à (C.matricule <> T[i-1].matricule)

T[i] ← C

Fin pour

Fin Analyse.

➤ **Analyse de la****procédure Transférer:**

DEF Proc Transferer(n : entier, T :

tab F : fiche)

Résultat : F rempli

Traitement : [recréer(F)]

Pour i de 1 à n faire

Ecrire(F, T[i])

Fin Pour

Avec T[i] faire

Si (Sport.Exercice = gymnastique)

Alors

Selon **Enchaînement** faire

A : moy ← formule1

B : moy ← formule2

C : moy ← formule3

D : moy ← formule4

Sinon

Ecrire("erreur dans le choix")

Fin selon

Fin si

Fin avec

Fin Analyse.

➤

Analyse de la**procédure Afficher:**

DEF Proc afficher (n : entier, T : tab)

Résultat :

Traitement : [nb ← 0]

Pour i de 1 à n faire

Avec T[i] faire

Si (Sport.Exercice = course) et (sport.note > 15) Alors

Nb ← nb + 1

Fin si

Fin Avec

Fin pour

Ecrire ("le nombre des candidats qui ont une note > 15 en course est : " nb)

Fin Analyse.

Remarque :

Dans la partie analyse la procédure saisir des champs d'un candidat n'est pas obligatoire.

Formules de Calcul des moyennes

Formule = (note1 + note2 + 2*note3) / 4

Avec note1, note2 et note3 sont respectivement les notes obtenues en course, L.poids et gymnastique.

Donc il faut écrire :

Si Sport.Exercice = Course Alors

Note1 $\leftarrow$ Sport.note

Fin si

Si Sport.Exercice = L.poids Alors

Note2 $\leftarrow$ Sport.note

Fin si

Si Sport.Exercice = Course Alors

Note3 $\leftarrow$ Sport.note

Fin si

➔ **Les formules 1, 2 et 3** sont basées sur l'enchaînement de la gymnastique choisie par le candidat :

Avec T[i] faire

Si Sport.Exercice = Gymnastique Alors

Note3 $\leftarrow$ Sport.note

Selon Enchaînement faire

A : Moy $\leftarrow$ (note 1 + note 2 + 1.5*note 3) / 3.5

B : Moy $\leftarrow$ (note 1 + note 2 + 2*note 3) / 4

C : Moy $\leftarrow$ (note 1 + note 2 + 2.5*note 3) / 4.5

D : Moy $\leftarrow$ (note 1 + note 2 + 3*note 3) / 5

Sinon

Ecrire ('erreur de calcul')

Fin selon

Fin si

Fin Avec.

