



Taki Academy
www.takiacademy.com

Algo & Prog

Classe : Bac Sciences de l'informatique

Série : Les Fichiers binaires(1)

Nom du Prof : Mr. Anis ELBEHI

📍 Sousse (Khezama - Sahloul) Nabeul / Sfax / Bardo / Menzah El Aouina /
Ezzahra / CUN / Bizerte / Gafsa / Kairouan / Medenine / Kébili / Monastir /
Gabes / Djerba / Jendouba / Sidi Bouzid / Siliana / Béja / Zaghouan



www.takiacademy.com



73.832.000



- Séance n° : 9
- Type : Théorique
- Objectif(s) :
 - Découvrir les fichiers binaires
 - Trier un fichier binaire
 - Revoir l'algorithme de tri par sélection

Rappel

Remarque

- Un fichier = une structure de données pouvant garder les données de façon permanente. On distingue deux types de fichiers : les fichiers texte et les fichiers de données (binaires)
- Un fichier possède deux noms : un nom logique et un nom physique
- Chaque fichier doit être enregistré sur un support de stockage physique (disque dur, flash, ...) et peut contenir une grande quantité d'informations.

Remarque

un fichier est comme un livre . il faut **l'ouvrir** pour **lire** , on peut prendre des notes (**écrire**) et il faut le **fermer** à la fin



Les fichiers de données

Algorithmique	Rôle
Ouvrir ("Chemin\Nom_physique", Nom_logique, "Mode")	Ouverture d'un fichier <ul style="list-style-type: none"> • Mode d'ouverture : <ul style="list-style-type: none"> ○ "rb" : Lecture (pointer au début) ○ "wb" : Ecriture (création) ○ "ab" : Ajout à la fin du fichier
Lire (Nom_logique, Objet)	Lecture d'un objet à partir d'un fichier
Ecrire (Nom_logique, Objet)	Écriture d'un objet dans un fichier
Fin_fichier (Nom_logique)	Retourne Vrai si le pointeur est à la fin du fichier sinon elle retourne Faux
Fermer (Nom_logique)	Fermeture d'un fichier

En algorithmique	En Python
Ouvrir ("Chemin\Nom_physique", Nom_logique , "Mode") Avec mode d'ouverture égal à : <ul style="list-style-type: none"> ○ "rb" : Lecture (pointer au début) ○ "wb" : Ecriture (création) ○ "ab" : Ajout à la fin du fichier 	Nom_logique= open ('Chemin\Nom_physique' , 'Mode')
Lire (Nom_logique , Objet)	from pickle import load , dump Objet = load (Nom_logique)
Ecrire (Nom_logique , Objet)	from pickle import load , dump dump (Objet , Nom_logique)
Fin_fichier (Nom_logique)	Fin_fichier = False while not (Fin_fichier) : try : x = load (Nom_logique) except : Fin_fichier = True
Fermer (Nom_logique)	Nom_logique. close ()

Exercice 1

🕒 90 min



- 1- Créer un fichier de données nommé physiquement « **liste.dat** » et logiquement **F**
- 2- Remplir le fichier F par les informations de N élèves sachant que $2 < N < 30$ et chaque élève est un enregistrement ayant les informations suivantes.

eleve		
Champ	Signification	Type et caractéristiques
id	Identifiant	Entier (<i>positif de 4 chiffres</i>)
np	Nom et prénom	Chaine de caractères (<i>ne peut contenir que des lettres alphabétiques minuscules et/ou des espaces</i>)
mg	Moyenne générale	Réel (<i>compris entre 0 et 20</i>)

- 3- Ensuite le programme doit afficher les élèves admis (nom, prénom et moyenne générale) par ordre de mérite (ordre décroissant basé sur la moyenne).

⚠ ATTENTION

- 1- Pour trier la liste des élèves, utiliser le tri par sélection.
- 2- Pour permuter le contenu de deux enregistrements E1 et E2 de type « eleve » vous pouvez faire appel à la procédure suivante :



Procédure permuter(@ E1, E2 : eleve)

Début

```
aux1 ← E1.id ; E1.id ← E2.id ; E2.id ← aux1
aux2 ← E1.np ; E1.np ← E2.np ; E2.np ← aux2
aux3 ← E1.mg ; E1.mg ← E2.mg ; E2.mg ← aux3
```

Fin

TDOL

Objet	Type / Nature
aux1	Entier
aux2	Chaine
aux3	réel

⚠ ATTENTION

Pour faire le tri du fichier il faut suivre les étapes suivantes :

- 1- Transférer le contenu du fichier vers un tableau d'enregistrements.
- 2- Trier le contenu du tableau T par ordre décroissant
- 3- Transférer le contenu du tableau trié vers le fichier.



Travail demandé :

- 1- Faire l'algorithme du programme principal
- 2- Faire les algorithmes des modules envisagés

1/ ALgorithme classement
 Debut
 $nph \leftarrow \text{"liste.dat"}$
 remplir(nph, N)
 trier(nph, N)
 afficher(nph, N)
 Fin

TDNT

types
elevé = enregistrement
id : entier
np : chaîne
mg : réel

Fin
 Fiche = Fichier d'élève
 TAB = Tableau de 29 élèves

TDDG

objet	T/N
nph	chaîne
N	entier
remplir	} procédure
trier	
afficher	

2/ procédure remplir(nph: chaîne, @ N: entier)

Debut

Repete
Lire(N)

Jusqu'à $2 < N < 30$

Ouvrir(nph, F, "wb")

pour i de 0 à N-1 Faire

Repete

Lire(E.id)

Jusqu'à $(1000 \leq E.id \leq 9999)$

Repete

Lire(E.np)

Jusqu'à Verif(E.np)

Repete

Lire(E.mg)

Jusqu'à $0 \leq E.mg \leq 20$

Ecrire(F, E)

Fin Pour

Fermer(F)

Fin

TDOL	
obj°	T/N
i	entier
E	eleve
F	Fiche
Verif	Fonction

Fonction Verif(ch: chaîne): booléen

Debut

$i \leftarrow 0$

test \leftarrow vrai

Tantque test et $i < \text{long}(ch)$ Faire

test $\leftarrow "a" \leq ch[i] \leq "z" \text{ ou } ch[i] = "_"$

$i \leftarrow i + 1$

Fin tantque

retourner test

Fin

TDOL	
obj°	T/N
i	entier
test	booléen

procédure Trier(nph :chaîne, N :entier)
 debut

transfert du Fichier vers un Tableau

Ouvrir(nph , F , "rb")

pour i de 0 à $N-1$ Faire

Lire(F , $T[i]$)

Fin Pour

Fermer(F)

trier Le Tableau d'enreg

pour i de 0 à $N-2$ Faire

$\left\{ \begin{array}{l} p \leftarrow i \\ \text{pour } j \text{ de } i+1 \text{ à } N-1 \text{ Faire} \\ \quad \text{si } T[j].mg > T[p].mg \text{ Alors} \\ \quad \quad p \leftarrow j \\ \text{Finsi} \\ \text{Fin Pour} \end{array} \right.$

si ($p \neq i$) Alors

Permuter ($T[i]$, $T[p]$)

Finsi

Fin Pour

transfert du Tableau vers le fichier

Ouvrir(nph , F , "wb")

pour i de 0 à $N-1$ Faire

Ecrire(F , $T[i]$)

Fin Pour

Fermer(F)

Fin

TDOL	
obj	T/N
i, j, p	entier
Permuter	procédure
F	Fiche
T	TAB

procedure afficher (nph: chaîne, N: entier)
Début

Ouvrir (nph, F, "rb")

pour i de 0 à N-1 Faire

Line(F, E)

Si E.mg \geq 10 Alors

Ecrire-nl(E.np, " ", E.mg)

Finsi

Fin Pour

Fermer (F)

Fin

$i+1$

TDOL

obj	T/N
i	entier
F	Fiche
E	elevé

! s'il demande
• d'afficher le rang