

Université de Bejaia	2 ^{ème} année licence RN	Nom :	Note :
Faculté des sciences exactes	Structure de fichier et structure de données	Prénom :	
Département Informatique	Examen session normale	Groupe :	

Q1 : 3 Pts

La taille (en octet) de la RAM installée sur une machine est de 5 GO. La taille des mots mémoire est de 4 octets.

1 - Quelle est la taille minimum du bus d'adresse de cette machine ?

2- Quelle la taille maximum en octet de la RAM que peut supporter cette machine ?

.....

1- La taille du bus d'adresse :

$$T = 5 * 2^{30} \text{ octets}$$

$$T = (5 * 2^{30}) / 4 \text{ mots mémoire}$$

$$T = 5 * 2^{28} \text{ mots}$$

la puissance de 2 pouvant contenir T est :

$$T = 2^3 * 2^{28}$$

$$T = 2^{31}$$

la taille minimum du bus d'adresse de cette machine est 31 bits

.....

2- La taille maximum de la RAM

$$T = 2^{31} \text{ mots mémoire}$$

$$T = 2^{31} * 4 \text{ octets}$$

$$T = 2^{33} \text{ octets}$$

$$T = 2^3 \text{ GO} = 8 \text{ GO}$$

.....

Q4 : 1 pt

Classez ces supports de stockage selon l'ordre croissant de leurs capacités de stockage.

- Disques optiques 2
- Disques SSD 1
- Disquettes 3

Q2 : 3 Pts

la taille occupée par un fichier de données sur le support de stockage est 2 GO. La taille des blocs physiques est de 512 octets. La taille des enregistrements logiques est de 128 octets. Quelle est la taille en octet du fichier logique si le facteur de blocage est égal à 2.

Nombre de bloc physique :

$$NB = 2 * 2^{30} / 2^9$$

$$NB = 2^{22}$$

Nombre d'enregistrement logique :

$$FB = \frac{NB \text{ enregistrement logique}}{NB \text{ bloc physique}}$$

$$NE = FB * NB$$

$$NE = 2 * 2^{22} = 2^{23}$$

Taille du fichier logique :

$$T = NE * 128 \text{ octet}$$

$$T = 2^{23} * 128 \text{ octet}$$

$$T = 2^{23} * 2^7 \text{ octet}$$

$$T = 2^{30} \text{ octet}$$

$$T = 1 \text{ GO}$$

.....

Q5 : 3 Pts

Soit une bande magnétique ayant ces caractéristiques : longueur de 400 m, des blocs physiques de 512 KO, espace inter-bloc de 16 mm, une densité d'enregistrement de 508 BPI (byte per Inch). (1 Inch = 2.54 cm).

1/ quelle est la capacité de stockage de la bande en octet ?

2/ étant un fichier de 1024 articles de 127 octets chacun, quelle est la longueur de bonde nécessaire pour stocker le fichier pour un facteur de blocage égale à 500.

1- capacité de stockage de la bande :

longueur de bloc

508 octet -----> 2.54 cm

512*1024 octet -----> LB

$LB = (512 * 1024 * 2.54) / 508 = 2621.44 \text{ cm} = 26.214 \text{ m}$

nombre de bloc

$NB = \text{longueur de bonde} / (LB + EIB)$

$NB = 400 \text{ m} / (26.214 \text{ m} + 0.016 \text{ m})$

$NB = 400 / 26.23 = 15.24$

NB = 16 blocs

taille = $16 * 512 \text{ KO} = 8 * 2 * 512 \text{ KO}$.

taille = 8 Mo

.....
.....
.....
.....

2- longueur de bandes nécessaire au fichier :

Nombre de bloc physique nécessaires au fichier :

$Nb = 1024 / 500 = 2.048 \rightarrow 3 \text{ blocs}$

longueur de bonde

$LB = 26.214 \text{ m}$, $EIB = 16 \text{ mm}$

$L = (LB + EIB) * Nb$

$L = (26.214 + 0.016) * 3 = 78.69 \text{ m}$

.....
.....
.....
.....
.....
.....
.....
.....

Exercice : (10 pts)

Nous souhaitons écrire un algorithme pour gérer les patients d'une clinique. Pour chaque patient on retient les informations suivantes : numéro (clé), nom, prénom, date d'admission. Les informations des patients sont sauvegardées dans un fichier avec un index primaire. Le fichier index contient des entrées sous forme de couples (numéro, adresse). L'index est trié selon l'ordre croissant des valeurs des numéros de patient. Les structures de données nécessaires sont définies comme suit :

type date = enregistrement

jour : 1..31 ;

mois : 1..12 ;

annee : entier ;

fin ;

patient = enregistrement;

numero : entier ;

nom, prenom : chaine[50] ;

date_admi : date ;

fin ;

entree = enregistrement ;

numero : entier ;

adresse : entier ;

fin ;

tab = tableau[1..1000] de entree ;

Travail à faire :

1- Définir la procédure **charger_index** permettant de charger le fichier index dans un tableau.

2- Ecrire une fonction **recherche** permettant de retrouver l'adresse d'un patient de clés C dans l'index, si le patient n'existe pas dans l'index la fonction renvoie 0.

2- Ecrire une procédure **insérer_index** pour insérer une nouvelle entrée dans l'index. La procédure doit maintenir le tableau index trié.

3- Ecrire une procédure **Ajouter** pour insérer un patient dans le fichier de donnée. La procédure doit vérifier l'existence du patient avant l'ajout. La procédure **Ajouter** doit insérer l'article dans le fichier de données et ajouter une entrée dans l'index.

Procedure charger_index(F: fichier, Var T:tab,Var n:entier) ;

Debut

n ← 0 ; ouvrir(F,lecture) ;

tantque(Non FdF(F))**faire**

2 pts

n ← n+1 ;

lire(F,T[n]) ;

finTantque ;

fermer(F) ;

Fin;

Fonction recherche (T:tab, n:entier, C : entier) ;

Var i: entier;

Debut

i←1 ;

tantque(C <= T[i].numero ET i<=n) **faire**

2 pts

i←i+1 ;

finTQ ;

Si (i<=n) **Alors**

retourner(T[i].adresse) ;

sinon

retourner(0) ;

finSi ;

fin ;

Procedure inserer_index(Var T:tab,Var n:entier, E :entree) ;

Var i,j: entree;

Debut

i ← 1 ;

tantque(E.numero > T[i].numero ET i<=n)**faire**

i ← i+1 ;

finTantque ;

n ← n+1 ; j ← n ;

2 pts

tantque(j>i) **faire**

T[j] ← T[j-1]; j ← j-1 ;

finTantque ;

T[i] ← E ;

Fin ;

fonction ajouter_patient(F :fichier de ouvrage, v : ouvrage) ;

Debut

Ouvrir(F,'ajout') ;

Ecrire(F,v) ;

1 pts

Retourner(position(F)) ;

```
Fin ;  
Procedure ajouter (F: fichier de ouvrage, T: tab, n :entier) ;  
Var E : entree ;  
    P : patient ;  
Debut  
    Lire(P.numero, P.nom, P.prenom) ;  
    Lire(P.date_admi.jour, P.date_admi.mois, P.date_admi.annee) ;  
    Si(recherche(T,n,P.numero)=0) Alors  
        Ecrire('le patient exixste deja') ;  
    Sinon  
        E.numero ← P.numero ;  
        E.adresse ← Ajouter_patient(F,P) ;  
        inserer_index(T,n,E) ;  
    finSi ;Fin ;
```

3 pts