Lycée : Houmt Souk
Matière : Algorithmique
Prof : Mr HAMDI Amor

Classe : 4 SI
Date : 20/10/2009
Durée : 1 heures

Devoir de contrôle théorique N°1

Position du problème et notation :

On veut écrire un programme de gestion de stock d'une pharmacie (drugstore).
Un médicament est caractérisé par:

1. Son code, exemple 4L276A (doit être unique)
2. Son nom, exemple aspegic1000
3. Sa nature, exemple liquide, comprimé, gélules…
4. Sa quantité, exemple 200
5. Son prix, exemple 1.980
6. Sa date de fabrication DF, exemple 20/09/2009
7. Sa date d'expiration DE, exemple 01/12/2011

Ce programme saisit une liste de N médicaments, N étant un nombre compris entre 1 et nmax, où nmax est une constante égale à 50, de telle sorte que les codes ne se répètent pas.

Le programme, selon le choix de l'utilisateur, permet les possibilités de chercher un médicament par son code, de chercher les médicaments périmés en demandant la date courante, d'afficher, sous forme tabulaire, le résultat de recherche ou toute la liste des médicaments.

Pour simplifier, le contrôle des données n'est imposé que sur les codes et les dates.

Travail demandé :

1. Analyser le problème en le décomposant en modules,
2. Analyser chacun des modules que vous avez proposés,
3. Déduire les algorithmes correspondant.

NB: une analyse comporte:

a) Le résultat
b) Le traitement
c) Les données (s'il y en a)
d) Le TDOU et le TDNT (s'il y en a)

Conseil: Soyez clairs, précis et concis dans votre rédaction!

Bon travail !

Citation du jour: «Bien écouter, c'est presque répondre". (Pierre Carlet de Marivaux)

Lycée : Houmt Souk **Classe : 4 SI**
Matière : Algorithmique **Date : 20/10/2008**
Prof : Mr HAMDI Amor **Durée : 1 heures**

Devoir de contrôle théorique N°1

Position du problème et notation :

On veut écrire un programme de gestion de stock d'une pharmacie (drugstore).
Un médicament est caractérisé par:

8. Son code, exemple 4L276A (doit être unique)
9. Son nom, exemple aspegic1000
10. Sa nature, exemple liquide, comprimé, gélules…
11. Sa quantité, exemple 200
12. Son prix, exemple 1.980
13. Sa date de fabrication DF, exemple 20/09/2009
14. Sa date d'expiration DE, exemple 01/12/2011

Ce programme saisit une liste de N médicaments, N étant un nombre compris entre 1 et nmax, où nmax est une constante égale à 50, de telle sorte que les codes ne se répètent pas.

Le programme, selon le choix de l'utilisateur, permet les possibilités de chercher un médicament par son code, de chercher les médicaments périmés en demandant la date courante, d'afficher, sous forme tabulaire, le résultat de recherche ou toute la liste des médicaments.

Pour simplifier, le contrôle des données n'est imposé que sur les codes et les dates.

Travail demandé :

1. Analyser le problème en le décomposant en modules,
2. Analyser chacun des modules que vous avez proposés,
3. Déduire les algorithmes correspondant.

NB: une analyse comporte:

e) Le résultat
f) Le traitement
g) Les données (s'il y en a)
h) Le TDOU et le TDNT (s'il y en a)

Conseil: Soyez clairs, précis et concis dans votre rédaction!

Bon travail !

Citation du jour: «Bien écouter, c'est presque répondre". (Pierre Carlet de Marivaux)

(Brouillon d'une solution ;))

PARTIE I : ANALYSE :

Analyse du Programme principal :

Résultat= Selon choix faire
1 : proc chercher_par_code(t,n)
2 : proc chercher_perime(t,n)
3 : proc afficher_tout(t,n)
Fin selon
Traitement= répéter
Choix=donnée ("Veuillez taper votre choix :
1 : pour chercher par code
2 : pour chercher périmé
3 : pour afficher tout ")

Jusqu'à choix dans [1..3]

Donnée= proc saisir (t, n)

T.D.O.U :

Objet	Type ou Nature	Rôle
Choix	Entier	Saisir le choix de l'utilisateur
N	Entier	Le nombre de médicaments à saisir
T	T_tab	Tableau d'enregistrements des médicaments
Nmax	Constante=50	Le nombre maximum des médicaments
Saisir	Procedure	Lecture des medicamets
chercher_par_code	Procedure	Chercher par code et afficher le resultat
chercher_perime	Procedure	Chercher les périmés et affiche le resultat
afficher_tout	Procedure	Affiche toute la liste des médicaments

T.D.N.T :

Type
T_tab= **tableau de nmax t_medicament** **T_medicament= enregistrement** **Code : chaine de caractères [6]** **Nom : chaine de caractères** **Nature : chaine de caractères** **Quantite: entier** **Prix : réel** **DF : chaine de caractères** **DE : chaine de caractères** **Fin t_medicament**

Analyse de la procédure saisir :

```
Def proc saisir(var t :t_tab ; var n :entier)
Resultat= T
Traitement :
        T=[] pour i de 1 à n faire
                Avec t[i] faire
                    Répéter
                            Code=donnée(”taper le code : ”)
                    Jusqu’à verif_code(code) et ((chercher(t,i-1,code)=0)ou(i=1))

                        Nom= donnée(”Taper le nom : ”)
                        Nature= donnée(”Taper la nature : ”)
                        Quantite= donnée(”Taper la quantité : ”)
                        prix= donnée(”Taper le prix : ”)

                        repeter
                            DF= donnée(”Taper la date de fabrication : ”)
                        Jusqu’à verif_date(DF)

                        répéter
                            DE= donnée(”Taper la date d’’expiration: ”)
                        Jusqu’à verif_date(DE)
                Fin avec
        Fin pour
Donnée :
        N=[] repéter
                N=Donnée(”Taper le nombre de médicaments :”)
            Jusqu’à (n>=1)et (n<=nmax)
```

T.D.O.U :

Objet	Type ou Nature	Rôle
i	Entier	Compteur
chercher	fonction	Vérifier si un code existe déjà ou non
verif_code	Fonction	Vérifier si un code est correct ou non
verif_date	Fonction	Vérifier si une date est correcte ou non

Analyse de la procédure chercher_par_code :

```
Def proc chercher_par_code(t :t_tab ; n :entier)
Resultat= R
Traitement :
    R=[] si chercher(t,X, n)=0 alors écrire(”Non trouvé”)
     Sinon Avec t[chercher(t,X, n)] faire
     Ecrire(code :8, nom :15,nature :15,quantite :4,prix :6 :3,DF :12,DE :12)
            FinAvec
     FinSi
 Donnée  X=donnée (”Taper le code du médicament à chercher : ”) ;
```

Objet	Type ou Nature	Rôle
i	Entier	Compteur
X	Chaine de caractère	Le code à chercher

Remarque : la fonction chercher doit être déclarée dans le programme principal !

Analyse de la procedure chercher_perimé

```
Def proc chercher_perime(t :t_tab ; n :entier)
Resultat= R
Traitement :
 R=[] pour i de 1 à n faire
    Avec t[i] faire
            Proc Extraire( DE,j1,m1,a1)
            Si (a2>a1) ou ((a2=a1)et(m2>m1))ou((a2=a1)et(m2=m1)et (j2>j1))
            Ecrire(code :8, nom :15,nature :15,quantite :4,prix :6 :3,DF :12,DE :12)
                  FinSi
           Fin Avec
    Fin Pour

    A2,m2,j2= proc extraire(date_courante,a2,m2,j2)
 Donnée :
            repeter
            Date_courante=donnée(”Taper la date courante :”) ;
            Jusqu’à verif_date(date_courante)
```

Objet	Type ou Nature	Rôle
i	Entier	Compteur
date_courante	Chaine de caractère	La date du jour
a1	entier	Année d’expiration
m1	entier	Mois d’expiration
j1	entier	Jour d’expiration
a2	entier	Année courante
m2	entier	Mois courant
j2	entier	Jour courant
extraire	procedure	Extraire le jour, le mois et l’année à partir d’une date donnée

Analyse de la procedure afficher_tout

```
Def proc afficher_tout(t :t_tab ; n :entier)
Resultat= R
Traitement :
R=[] Ecrire(”code” :8, ”nom” :15, ”nature” :15, ”qtite” :4, ”prix” :6 , ”DF” :12, ”DE” :12)
```

Pour i de 1 à n faire avec t[i] faire
Ecrire(code :8, nom :15,nature :15,quantite :4,prix :6 :3,DF :12,DE :12)
Fin Avec
Fin Pour

TDOU

Objet	Type ou Nature	Rôle
i	Entier	compteur

PARTIE II : LES ALGORITHMES :

Programme principal :

0. Debut Drugstore
1. proc saisir (t, n)

2. répéter
   Ecrire ("Veuillez taper votre choix : ")
   Ecrire (" 1 : pour chercher par code")
   Ecrire (" 2 : pour chercher périmé")
   Ecrire (" 3 : pour afficher tout ")

   Lire(choix)

   Jusqu'à choix dans [1..3]

3. Selon choix faire
   1 : proc chercher_par_code(t,n)
   2 : proc chercher_perime(t,n)
   3 : proc afficher_tout(t,n)
   Fin selon
4. Fin drugstore

La procédure saisie :

0. Def proc saisir(var t :t_tab ; var n :entier)
1. repéter
   Ecrire("Taper le nombre de médicaments :")
   Lire(N)
   Jusqu'à (n>=1)et (n<=nmax)

2. pour i de 1 à n faire
   Avec t[i] faire
   Répéter
   Ecrire("taper le code : ")
   Lire(code)
   Jusqu'à verif_code(code) et ((chercher(t,i-1,code)=0)ou(i=1))

   Ecrire("Taper le nom : ") Lire(nom)
   Ecrire("Taper la nature : ") lire(nature)
   Ecrire("Taper la quantité : ") lire(quantite)

Ecrire(”Taper le prix : ”) lire(prix)

repeter
Ecrire(”Taper la date de fabrication : ”)
Lire(DF)
Jusqu’à verif_date(DF)

répéter
Ecrire(”Taper la date d’’expiration: ”)
Lire(DE)
Jusqu’à verif_date(DE)
Fin avec
Fin pour

3. **Fin** saisir

La procedure chercher_par_code :

0. **Def proc chercher_par_code(t :t_tab ; n :entier)**
1. **Ecrire(**”Taper le code du médicament à chercher : ”) lire(X)
2. si **chercher**(t,X, n)=0 alors écrire(”Non trouvé”)
   Sinon **Avec** t[chercher(t,X, n)] faire
   Ecrire(code :8, nom :15,nature :15,quantite :4,prix :6 :3,DF :12,DE :12)
   FinAvec
   FinSi
3. **Fin chercher_par_code**

la procedure chercher_perimé

0. **Def proc chercher_perime(t :t_tab ; n :entier)**
1. **repeter**
   Ecrire(”Taper la date courante :”) ,
   Lire(Date_courante)
   Jusqu’à verif_date(date_courante)

2. proc extraire(date_courante,a2,m2,j2)

3. pour i de 1 à n faire
   Avec t[i] faire
   Proc Extraire(DE,a1,m1,j1)
   Si (a2>a1) ou ((a2=a1)et(m2>m1))ou((a2=a1)et(m2=m1)et (j2>j1))
   Ecrire(code :8, nom :15,nature :15,quantite :4,prix :6 :3,DF :12,DE :12)
   FinSi
   Fin Avec
   Fin Pour

4. **Fin chercher_perimé**

Algorithme de la procédure afficher_tout

```
0.Def proc afficher_tout(t :t_tab ; n :entier)
1. Ecrire("code" :8, "nom" :15, "nature" :15, "qtite" :4, "prix" :6 , "DF" :12, "DE" :12)
  2.Pour i de 1 à n faire avec t[i] faire
     Ecrire(code :8, nom :15,nature :15,quantite :4,prix :6 :3,DF :12,DE :12)
      Fin Avec
   Fin Pour
   4.  Fin afficher_tout
```

Algorithme de la procedure extraire :

```
0.Def proc extraire(D :chaine ; var a,m,j :entier)
1.valeur(sous-chaine(d,1,2),j,pe)
2.valeur(sous-chaine(d,4,2),m,pe)
3.valeur(sous-chaine(d,7,4),a,pe)
4.fin extraire
```

Algorithme de la fonction verif_date

```
0.def Fn verif_date(d :chaine de caractere) :booléen
1.valeur( sous_chaine(d,1,2), j,pe1)
2.valeur(sous_chaine(d,4,2), m,pe2)
3.valeur(sous_chaine(d,7,4), m,pe3)
4. verif_date ← (j dans [1..31])et (m dans [1..12])et(a>=1900)et(long(d)=10)
                Et (pe1 =0) Et (pe2 =0) Et (pe3 =0) et (d[3]='/')et (d[6]='/')
5.finverif_date
```

Algoritme de la fonction verif_code

```
0.def fn verif_code(ch :string) :booléen
1. Valeur(sous-chaine(ch,3,3),N,pe)
2. verif_code ←(ch[1] dans ['0'..'9']) et (ch[2] dans['A'..'Z'])et (ch[5] dans['A..'Z'])
   et(pe=0)
3.fin verif_code
```

Algorithme de la fonction chercher

```
0.Def Fn recherche(t :t_tab ; X :chaine ; n :entier) :entier
1.i←0
   Repeter
         i←i+1
   jusqu'à (t[i].code= X)ou (i=n)
2. si ← (t[i].code= X) alors chercher←i  sinon chercher←0  FinSi
3.fin chercher
```

Citation du jour: «Bien écouter, c'est presque répondre". (Pierre Carlet de Marivaux)