**Lycée  : Houmt Souk Classe : 4 SI**

**Matière : Algorithmique Date : 20/10/2009**

**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:

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:

1. Le résultat
2. Le traitement
3. Les données (s'il y en a)
4. Le TDOU et le TDNT (s'il y en a)

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

**Bon travail !**

**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:

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:

1. Le résultat
2. Le traitement
3. Les données (s'il y en a)
4. Le TDOU et le TDNT (s'il y en a)

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

**Bon travail !**

**(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**  **N**  **T**  **Nmax**  **Saisir**  **chercher\_par\_code**  **chercher\_perime**  **afficher\_tout** | Entier  Entier  T\_tab  Constante=50  Procedure  Procedure  Procedure  Procedure | Saisir le choix de l’utilisateur  Le nombre de médicaments à saisir  Tableau d’enregistrements des médicaments  Le nombre maximum des médicaments  Lecture des medicamets  Chercher par code et afficher le resultat  Chercher les périmés et affiche le resultat  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**  **chercher**  **verif\_code**  **verif\_date** | Entier  fonction  Fonction  Fonction | Compteur  Vérifier si un code existe déjà ou non  Vérifier si un code est correct ou non  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**  **X** | Entier  Chaine de caractère | Compteur  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**  **date\_courante**  **a1**  **m1**  **j1**  **a2**  **m2**  **j2**  **extraire** | Entier  Chaine de caractère  entier  entier  entier  entier  entier  entier  procedure | Compteur  La date du jour  Année d’expiration  Mois d’expiration  Jour d’expiration  Année courante  Mois courant  Jour courant  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 :**

1. Debut Drugstore
2. proc saisir (t, n)
3. 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]

1. Selon choix faire

1 : proc chercher\_par\_code(t,n)

2 : proc chercher\_perime(t,n)

3 : proc afficher\_tout(t,n)

Fin selon

1. Fin drugstore

**La procédure saisie :**

1. Def proc saisir(var t :t\_tab ; var n :entier)
2. repéter

Ecrire(″Taper le nombre de médicaments :″)

Lire(N)

Jusqu’à (n>=1)et (n<=nmax)

1. 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

1. **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

1. **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**