| République Tunisienne<br>MINISTÈRE DE L'EDUCATION<br>Commissariat Régional de Mahdia | *Section :* Math, Sc. Exp. & Tech. | |
|---|---|---|
| | *Matière :* Informatique | |
| *Niveau :* 4ème Année | *Date :* Mardi 15 Mai 2018 | *Durée :* 1 Heure 30' |

# Devoir de Synthèse N° 2

Nom & Prénom : ..... Classe : ..... Note : ..... /20

Remise du devoir sera le 21/05/2018 à 10h

*N.B. : Le sujet comporte 2 pages à remettre à la fin de l'épreuve*

## Exercice 1 : (3 points)

Soit le tableau de déclaration des objets suivant :

| T.D.O. | |
|---|---|
| **Objet** | **Nature/Type** |
| C | Var / Caractère |
| S | Var / Caractère |
| R | Var / Entier |

Compléter le tableau ci-dessous par l'instruction d'affectation correspondante à chacune des tâches à faire suivantes :

| Tâche à faire | Instruction d'affectation |
|---|---|
| Affecter dans la variable C, une lettre Majuscule au hasard. | C <-- CHR (65 + Aléa (26)) ou c<--chr(ord("A")+aléa(ord("Z")-ord("A")+1)) |
| Convertir la lettre de la variable C en lettre minuscule. | INC (C, 32) ou C <-- CHR(ORD(C)+32) |
| Affecter dans la variable R, le rang dans l'alphabet Français de la lettre contenue dans la variable C. | R <-- ORD(C) - ORD("a")+1 |
| Affecter dans la variable S, le caractère symétrie de C, sachant que la variable C contient déjà une lettre en minuscule. | S <-- CHR (ORD("a")+ORD("z")-ORD(C)) |

## Exercice 2 : (3 points)

Soit le tableau de déclaration des nouveaux types suivant :

| Type |
|---|
| Tab = Tableau de 100 entiers |
| Mois = (Avril, Mai, Juin) |

Transformer chacune des entêtes de ces procédures en un entête de fonction si cela est possible, sinon écrire « **Impossible** ».

| Entête d'une Procédure | Entête d'une Fonction |
|---|---|
| DEF PROC Proc1 (Ch1 : Chaine; Var Ch2 : Chaine) ; | DEF FN Fonct1(ch1:chaine):chaine |
| DEF PROC Proc2 (Var R : Tab ;T : Tab ; X : Entier) ; | Impossible |
| DEF PROC Proc3 (T : Tab ; X : Entier; Var M : Mois) ; | DEF FN Fonct3(T:tab; X:entier):Mois |
| DEF PROC Proc4 (Var Test : Booléen) ; | DEF FN Fonct4 : Booléen |

**Problème : (14 Points)**

Pour recharger un téléphone portable, un client doit composer *XYZ*code de recharge#

Un code de recharge est valide s'il vérifie les contraintes suivantes :

- Il est composé de **13 chiffres**.
- Le nombre formé par **les 3 premiers chiffres** est un **nombre premier**.
- Il contient **au moins une séquence de quatre chiffres** qui forment **les termes successifs d'une suite arithmétique** (la différence entre 2 chiffres successifs soit la même).

***Exemple de validité de code :***

Soit l'essai de recharge suivante : *XYZ*1376357950241#

➔ Le code C = 1376357950241

- Le code C saisi contient 13 chiffres
- Le nombre formé par les 3 premiers chiffres (137) est un nombre premier
- Le code contient une séquence de 4 chiffres (3579) qui forment les termes successifs d'une suite arithmétique de raison 2.

➔ Ce code **C** est donc **valide**.

Une fois le code **C** est valide, on passe à la vérification de son existence dans le tableau **Code** et de sa disponibilité dans le tableau **Etat** sachant que :

- **Code : Tableau de n Chaînes ($10 \leq n \leq 100$)** contenant les n codes de recharge.
- **Etat : Tableau de Caractères** qui peut contenir soit la lettre '**U**' si le code est Utilisé ou la lettre '**L**' si le code est encore **L**ibre.

On se propose d'écrire un programme permettant :

- de remplir le tableau **Code** par des chaînes de 13 chiffres selon les contraintes déjà décrites ci-dessus.
- d'initialiser tous les éléments du tableau **Etat** à '**L**' c'est à dire que tous les codes du tableau sont au départ libres.
- de saisir un **code de recharge C** selon les contraintes déjà décrites ci-dessus.
- de vérifier l'existence du code de recharge saisi dans le tableau **Code** et de sa disponibilité dans le tableau **Etat** et d'afficher l'un des messages suivants :
  - « **Code invalide** » : Si le code n'est pas valide.
  - « **Code inexistant** » : Si le code est valide mais il n'existe pas dans le tableau **Code**
  - « **Code utilisé** » : Si le code est valide mais il est déjà utilisé.
  - « **Opération de recharge réussite** » : Si le code est **valide** et il est encore **libre**, dans ce cas, **on doit mettre à jour l'état du code qui change de 'L' à 'U'** (afin de ne pas pouvoir l'utiliser une deuxième fois).

**Travail demandé :**

1) Analyser le problème en le décomposant en modules.
2) Ecrire les algorithmes et les tableaux de déclaration relatifs aux modules envisagés.

## Analyse du P.P. : (2.75 pts)

**Nom : Recharge**

```
Résultat = [  C = Donnée ("Saisir le code de recharge")  ]                    (0.25 pt)
        Si  Non FN PREMIER(C) ET Non FN SUITE(C) ET (Long(C)≠13)              (0.5 pt)
            Alors Ecrire ("Code invalide")                                    (0.25 pt)
            Sinon Si FN POSITION(C, CODE, N) = 0                              (0.25 pt)
                    Alors Ecrire ("Code inexistant")                          (0.25 pt)
                    Sinon Si  ETAT[FN POSITION(C,CODE,N)]="U"                 (0.25 pt)
                            Alors Ecrire ("Code utilisé")                     (0.25 pt)
                            Sinon Ecrire("Opération de recharge réussite")    (0.25 pt)
                                  ETAT[FN POSITION(C,CODE,N)] ← "U"           (0.25 pt)
        Fin Si

(N, CODE, ETAT) = PROC SAISIES(N, CODE, ETAT)                                 (0.25 pt)

FIN Recharge
```

**Tableau de Déclaration des Nouveaux Types** (0.5 pt)

| Type |
|---|
| Tab1 = Tableau de 100 chaînes |
| Tab2 = Tableau de 100 caractères |

**Tableau de Déclaration des Objets globaux** (0.5 pt)

| Objets | Type/Nature | Rôle |
|---|---|---|
| Code | Tab1 | |
| Etat | Tab2 | |
| N | Entier | |
| C | Chaîne | |
| Saisies | Procédure | |
| Suite | Fonction | |
| Premier | Fonction | |
| Position | Fonction | |

## ALGORITHME DE LA PROCEDURE SAISIES : (4 pts)

```
0)  DEF PROC SAISIES (Var n:Entier ; Var code:tab1 ; Var etat:tab2)          (0.75 pt)
1)  Répéter
        Ecrire ("N = "), Lire (N)                                           (0.5 pt)
    Jusqu'à  n  dans [10..100]                                              (0.5 pt)
2)  Pour  i  de  1  à  n  Faire                                             (0.5 pt)
        Répéter
            Lire (code[i])                                                  (0.5 pt)
        Jusqu'à  FN premier(code[i]) et FN suite(code[i]) et (Long(code[i])=13)  (0.75 pt)
        etat[i] ← "L"                                                       (0.5 pt)
    Fin Pour
3)  Fin Saisies
```

**Tableau de Déclaration des Objets locaux**

| Objets | Type/Nature | Rôle |
|---|---|---|
| i | Entier | Compteur |

## ALGORITHME DE LA FONCTION PREMIER : (2 pt)

```
0) DEF FN PREMIER (ch : chaîne) : Booléen          (0.25 pt)
1) ch1 ← Sous_chaîne (ch,1,3)                      (0.25 pt)
2) Valeur (ch1, x, err)                            (0.25 pt)
3) d ← 2                                           (0.25 pt)
4) Pour  i  de  2  à  (x Div 2)  Faire             (0.25 pt)
        Si  (x Mod i) = 0                          (0.25 pt)
            Alors  d ← d+1                         (0.25 pt)
        Fin si
   Fin pour
5) premier ← d=2                                   (0.25 pt)
6) Fin Premier
```

**Tableau de Déclaration des Objets locaux** (0.5 pt)

| Objets | Type/Nature | Rôle |
|---|---|---|
| x, err, d, i | Entier | |
| ch1 | Chaîne | |

## ALGORITHME DE LA FONCTION SUITE : (1.5 pt)

```
0) DEF FN SUITE (ch : chaîne) : Booléen
1) Valeur (ch[1], x1, err)
2) Valeur (ch[2], x2, err)
3) r ← x2-x1
4) nb ← 2
5) i ← 3
6) Répéter
      x1 ← x2
      Valeur (ch[i], x2, err)
      Si  x2 = x1 + r   Alors   nb ← nb+1
                        Sinon   nb ← 2
                                r ← x2-x1
      fin si
      i ← i+1
   Jusqu'à   (i > Long(ch)) Ou (nb ≥ 4)
7) suite ← nb ≥ 4
8) Fin Suite
```

**Tableau de Déclaration des Objets locaux** (0.5 pt)

| Objets | Type/Nature | Rôle |
|---|---|---|
| i, x1, x2, err, r, nb | Entier | |

## ALGORITHME DE LA FONCTION POSITION : (1.75 pt)

0) **DEF FN POSITION (c : chaîne ; code : tab1 ; n : entier) : Entier** **(0.25 pt)**
1) **position ← 0** **(0.25 pt)**
2) **i ← 0** **(0.25 pt)**
3) **Répéter**
   **i ← i+1** **(0.25 pt)**
   **Jusqu'à (c = code[i]) Ou (i=n)** **(0.25 pt)**
4) **Si c= code[i]** **(0.25 pt)**
   **Alors position ← i** **(0.25 pt)**
   **Fin si**
5) **Fin position**

**Tableau de Déclaration des Objets locaux**

| Objets | Type/Nature | Rôle |
|---|---|---|
| i | Entier | compteur |

```
Program bacblanc2018;
Uses Wincrt;
Type
  tab1 = Array[1..100] Of String;
  tab2 = Array[1..100] Of Char;
Var
  code: tab1;
  etat: tab2;
  n: Integer;
  c: String;
 (*****************************************************************)
Function premier(ch:String): Boolean;
Var
  ch1: String;
  x,err,d,i: Integer;
Begin
  ch1 := Copy(ch,1,3);
  Val (ch1,x,err);
  d := 2;
  For i:=2 To x Div 2 Do
    If x Mod i =0
      Then d := d+1;
  premier := d=2;
End;
 (*****************************************************************)
Function suite(ch:String): Boolean;
Var
  i,x1,x2,err,r,nb: Integer;
Begin
  Val(ch[1],x1,err);
  Val(ch[2],x2,err);
  r := x2-x1;
  i := 3;
  nb := 2;
  Repeat
    x1 := x2;
    Val(ch[i],x2,err);
    If x2=x1+r
      Then nb := nb+1
    Else
      Begin
        nb := 2;
        r := x2-x1;
      End;
    i := i+1;
  Until (i>Length(ch)) Or (nb>=4);
  suite := nb>=4;
End;
(*****************************************************************)
```

```
Procedure saisies(Var n:Integer;Var code:tab1 ;Var etat:tab2);
Var
  i: Integer;
Begin
  Repeat
    Write('n = ');
    Readln(n);
  Until n In [10..100];
  For i:=1 To n Do
    Begin
      Repeat
        Readln(code[i]);
      Until premier(code[i]) And suite(code[i]) And (Length(code[i])=13);
      etat[i] := 'L';
    End;
End;
(*************************************************************************)
Function position(c:String;code:tab1;n:Integer): Integer;
Var
  i: Integer;
Begin
  position := 0;
  i := 0;
  Repeat
    i := i+1;
  Until (c= code[i])Or (i=n);
  If c= code[i] Then position := i;
End;
 (*************************P.P*****************************)
Begin
  saisies(n,code,etat);
  Writeln('saisir code de recharge');
  Readln(c);
  If Not premier(c) And Not suite(c) And (Length(c)<>13)
    Then Writeln('code invalide')
  Else If position(c,code,n)=0
         Then Writeln('code inexistant')
  Else If etat[position(c,code,n)]='U'
         Then Writeln('code utilisé')
  Else
    Begin
      Writeln('opération de recharge réussite');
      etat[position(c,code,n)] := 'U' ;
    End;
End.
```