EXAMEN DU BACCALAURÉAT SESSION 2015

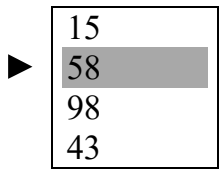
Corrigé Session principale

Épreuve : ALGORITHMIQUE ET PROGRAMMATION

Exercice 1 : *(10,5 points : 0,5 x 15 + 1 x 3)*

Dans un contexte informatique et pour chacune des propositions données ci-dessous, mettre dans chaque case, la lettre **V** si la proposition est correcte, ou la lettre **F** dans le cas contraire :

1) Soit un fichier d'entiers, ayant comme nom logique **F.** On suppose que le pointeur est positionné sur le deuxième entier comme indiqué ci-dessous.



N.B : Toutes les instructions données dans a), b) et c) sont valides.

a) Le contenu de X après l'exécution de l'instruction **Lire (F, X)** est :

[F] 15 [V] 58 [F] 98

b) L'instruction **Pointer (F, 3)** permet de positionner le pointeur sur l'enregistrement ayant la valeur :

[F] 58 [F] 98 [V] 43

c) Le contenu de Y après l'exécution de l'instruction **Y ← Fin_fichier (F)** est :

[F] 43 [F] 4 [V] faux

2) Soit la suite U définie par : $\begin{cases} U_0 = 1 \\ U_n = 2*U_{n-1} + n \end{cases}$ avec n un entier supérieur ou égal à 1

a) U est une suite récurrente d'ordre :

[V] 1 [F] 2 [F] 5

b) Le 3ème terme de la suite U (U_2) est égal à :

[F] 5 [V] 8 [F] 9

c) L'algorithme permettant de calculer le $n^{ième}$ terme de la suite **U** (avec n > 1) est :

F

```
0) Def fn terme (n : entier) : entier
1) t[0] ← 1
2) Pour i de 1 à n faire
       t[i] ← 2*t[i-1]+n
   Fin pour
3) terme ← t[n]
4) Fin terme
```

V

```
0) Def fn terme (n : entier) : entier
1) Si n=0 alors terme ← 1
     Sinon
        terme ← 2*terme(n-1)+n
     Fin si
   Fin si
2) Fin terme
```

F

```
0) Def fn terme (n : entier) : entier
1) Up ← 1
2) Pour i de 2 à n faire
       Up ← 2*Up+i
   Fin pour
3) terme ← Up
4) Fin terme
```

Exercice 2 : (6 points)

```
DEF FN Verif_auto_nombre (N : entier) : Booléen
Résultat = Verif_auto_nombre ← n <> S+ i
        (S, i) = [i ← 0]
        Répéter
                i ← i + 1
                M ← i
                S ← 0
                Répéter
                        S ← S + M mod 10
                        M ← M div 10
                Jusqu'à (M =0)
        Jusqu'à (n = i + S) ou (i = n-1)
```

TDO	
Objet	Type
i	Entier
M	Entier
S	Entier

Exercice 3 : (6 points)

```
0) DEF FN Longueur_Max(epsilon, p:réel):réel
1) Lopt ← 0
   Smax ← 0
   Répéter
        Lopt ← Lopt+epsilon
        S ← Smax
        Smax ← Lopt*(p-2*Lopt)/Pi
   Jusqu'à ((Smax - S) <= 0) ou (Lopt >=p/2)
2) Longueur_Max ← Lopt - epsilon
3) Fin FN Longueur_Max
```

Problème : (17,5 points)

Analyse du programme principal :

Résultat = MDPF
MDPF =
    Associer (F_FORT,"c:\MDPFort.txt")
    Proc Generer_F_FORT (F_FORCE, F_FORT)
F_FORCE =
    Associer (F_FORCE,"c:\forceMDP.DAT")
    Proc Generer_F_FORCE (MP, F_FORCE)
MP =
    Associer (MP, "c:\Motspass.txt")

Tableau de déclaration de nouveaux types
ENRG = Enregistrement Mpass, Force : chaîne de caractères Score : entier Fin ENRG
F = Fichier de ENRG

Tableau de déclaration des objets globaux

Objet	Type/Nature
F_FORCE	F
F_FORT	Texte
MP	Texte
Generer_F_FORT	Procédure
Generer_F_FORCE	Procédure

Analyse de la procédure Generer_F_FORT:

Def Proc Generer_F_FORT (var F_FORCE : F; var F_FORT : Texte)
Résultat = F_FORT
F_FORT =
    [Ouvrir (F_FORCE), Récréer (F_FORT)]
    Tantque non Fin_fichier (F_FORCE) Faire
        Lire (F_FORCE, e)
        Si e.force = "Très fort" Alors
            Ecrire_nl (F_FORT, e.Mpass)
        FinSI
    FinTantque
    Ecrire_nl (F_FORT)
    [Ouvrir (F_FORCE)]
    Tantque non Fin_fichier (F_FORCE) Faire
        Lire(F_FORCE, e)
        Si e.force = "Fort" Alors
            Ecrire_nl (F_FORT, e. Mpass)
        FinSI
    FinTantque
    Fermer (F_FORCE)
    Fermer (F_FORT)

T.D.O.L	
Objet	**Type/Nature**
e	ENRG

Analyse de la procédure Generer_F_FORCE :

```
Def Proc Generer_F_FORCE (var MP : Texte ; var F_FORCE : F)
Résultat = F_FORCE
F_FORCE = [Recréer (F_FORCE),    Ouvrir(MP)]
          Tantque non Fin_fichier (MP) Faire
              Lire_nl (MP, ch)
              r.Mpass ← ch
              Proc Nb_Min_Maj (ch, min, maj)
              r.score ← long (ch)*4 + (long (ch) – min) * 3 + (long (ch) - maj) * 2 + (long (ch) -
                    (min + maj)) * 5 – FN SeqMax (ch , "a", "z") * 2 – FN SeqMax (ch, "A", "Z") * 2
              Si r.score < 20 Alors r.force ←"Très faible"
              Sinon Si r.score < 40 Alors r.force ← "Faible"
              Sinon Si r.score < 60 Alors r.force ← "Moyen"
              Sinon Si r.score < 80 Alors r.force ← "Fort"
              Sinon r.force ← "Très fort"
              FinSi
              Ecrire (F_FORCE, r)
            FinTantque
         Fermer (F_FORCE)
         Fermer (MP)
```

T.D.O.L	
Objet	**Type/Nature**
Ch	Chaîne
r	ENRG
min	Entier
max	Entier
SeqMax	Fonction
Nb_Min_Maj	Procédure

Analyse de la procédure Nb_Min_Maj:

```
Def Proc Nb_Min_Maj (ch : Chaine ; Var min, maj : Entier)
Résultat = min, maj
(min, maj) = [ min←0 ; maj←0]
              Pour i de 1 à long (ch) Faire
               Si ch [i] dans ["a".."z"] Alors
                        min ← min+1
               FinSi
               Si ch [i] dans ["A".."Z"] Alors
                        maj ← maj+1
               FinSi
             FinPour
```

T.D.O.L	
Objet	**Type/Nature**
i	Entier

Analyse de la fonction SeqMax:

```
Def FN SeqMax (ch : Chaine ; c1, c2 : Caractère) : Entier
Résultat = SeqMax ← Min
Min = [ncour ←0; npred ←0]
       Pour i de 1 à long (ch) Faire
         Si ch[i] dans [c1..c2] Alors
              ncour ← ncour+1
        Sinon
              Si npred < ncour Alors
                    npred ← ncour
              FinSi
              Ncour ← 0
         FinSI
      FinPour
      Si npred < ncour Alors  Min ← ncour
      Sinon  Min ← npred
      FinSi
```

T.D.O.L	
Objet	**Type/Nature**
Min, ncour, npred, i	Entier