

# EXAMEN DU BACCALAURÉAT SESSION 2015

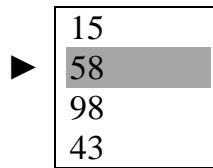
## Corrigé Session principale

### 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 :

15

58

98

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

58

98

43

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

43

4

faux

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

a) U est une suite récurrente d'ordre :

1

2

5

b) Le 3<sup>ème</sup> terme de la suite U ( $U_2$ ) est égal à :

5

8

9

c) L'algorithme permettant de calculer le n<sup>ème</sup> terme de la suite U (avec  $n > 1$ ) est :

- |   |  |   |
|---|--|---|
| <b>F</b> 0) Def fn terme (n : entier) : entier<br>1) t[0] ← 1<br>2) Pour i de 1 à n faire<br>t[i] ← 2*t[i-1]+n<br>Fin pour<br>3) terme ← t[n]<br>4) Fin terme | <b>V</b> 0) Def fn terme (n : entier) : entier<br>1) Si n=0 alors terme ← 1<br>Sinon<br>terme ← 2*terme(n-1)+n<br>Fin si<br>Fin si<br>2) Fin terme | <b>F</b> 0) Def fn terme (n : entier) : entier<br>1) Up ← 1<br>2) Pour i de 2 à n faire<br>Up ← 2*Up+i<br>Fin pour<br>3) terme ← Up<br>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<br>Mpass, Force : chaîne de caractères<br>Score : entier<br>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 : Chaîne ; 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 : Chaîne ; 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      |