

# **Corrigé Informatique Sections : Mathématiques, Sciences expérimentales, Sciences Techniques**

## **Session principale Baccalauréat 2017**

### **Exercice N°1 :**

Réécrire dans la colonne « **Correction** », les phrases données dans la colonne « **Proposition** » en apportant les modifications nécessaires afin qu'elles soient correctes :

| <b>Proposition</b>                                                                                                               | <b>Correction</b>                                                                                                                                                                                                                                                                                                    |
|----------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| La recherche dichotomique est une méthode de recherche appliquée sur un tableau quelconque.                                      | <ul style="list-style-type: none"> <li>• La recherche dichotomique est une méthode de recherche appliquée sur un tableau <b>trié</b>.</li> <li>• La recherche <b>séquentielle</b> est une méthode de recherche appliquée sur un tableau quelconque.</li> </ul>                                                       |
| Un tableau est une structure de données permettant de regrouper plusieurs éléments de types différents.                          | Un tableau est une structure de données permettant de regrouper plusieurs éléments de <b>même</b> type.                                                                                                                                                                                                              |
| Les paramètres formels sont utilisés lors de l'appel d'un sous-programme.                                                        | <ul style="list-style-type: none"> <li>• Les paramètres formels sont utilisés lors de <b>la définition</b> d'un sous-programme (<b>au niveau de l'entête</b>).</li> <li>• Les paramètres <b>effectifs</b> sont utilisés lors de l'appel d'un sous programme.</li> </ul>                                              |
| L'évaluation d'une expression formée par des opérateurs ayant la même priorité se fait de gauche à droite ou de droite à gauche. | L'évaluation d'une expression formée par des opérateurs ayant la même priorité se fait de <b>gauche à droite</b> .                                                                                                                                                                                                   |
| Le type scalaire énuméré définit un ensemble ordonné et fini de valeurs appartenant à un type prédéfini.                         | <ul style="list-style-type: none"> <li>• Le type scalaire énuméré définit un ensemble ordonné et fini de valeurs <b>n'appartenant pas à un type prédéfini dites identificateurs</b>.</li> <li>• Le <b>type intervalle</b> définit un ensemble ordonné et fini de valeurs appartenant à un type prédéfini.</li> </ul> |

**Exercice 2 :**

- a) Tableau de déclaration des objets de l'algorithme « Exercice »

**T.D.O**

| Objet | Type/nature | Rôle                        |
|-------|-------------|-----------------------------|
| ch1   | Chaîne      | La première chaîne à saisir |
| ch2   | Chaîne      | La deuxième chaîne à saisir |
| p     | Entier      | Position                    |
| ch    | Chaîne      | Chaîne intermédiaire        |
| i     | Entier      | Compteur                    |

- b) expression équivalente utilisant un ou plusieurs modules prédéfinis

Séquence n°2 :  $ch \leftarrow \text{Sous-chaîne}(ch1, 1, p-1)$ Séquence n°4 :  $ch \leftarrow \text{Concat}(ch, \text{Sous-chaîne}(ch1, p, \text{Long}(ch1)-p+1))$ 

- c) Le programme affiche : BAC 2016/2017
- d) Cet algorithme permet d'insérer une chaîne ch2 dans une chaîne ch1 à une position p.
- e) le module prédéfini qui peut jouer le même rôle est : Insère(ch2, ch1, p)

**Problème :****1) Analyse du programme principal**

Nom : Hitparade

Résultat= PROC Afficher (T1, T2)

(T1,T2)= [PROC Chanson (T1)

PROC Score (T2,N)]

PROC Trier (T1,T2)

N= PROC Saisir(N)

Fin Hitparade

**TDNT**

| Type                       |
|----------------------------|
| Tab1= tableau de 5 chaînes |
| Tab2= tableau de 5 entiers |

**TDOG**

| Nom      | Nature/Type | Rôle                              |
|----------|-------------|-----------------------------------|
| T1       | Tab1        | Tableau des chansons              |
| T2       | Tab2        | Tableau des scores                |
| N        | Entier      | Nombre de participants            |
| Saisir   | Procédure   | Saisir le nombre des participants |
| Chanson  | Procédure   | Remplir le tableau des chansons   |
| Score    | Procédure   | Comptabiliser les scores          |
| Trier    | Procédure   | Trier les scores et les chansons  |
| Afficher | Procédure   | Afficher le résultat              |

**2) Les algorithmes des modules envisagés :**

**a) Module Chanson :**

0) **DEF PROC Chanson (Var T1 : Tab1)**

1) Pour i de 1 à 5 faire

    Répéter

        Ecrire (" Saisir le titre de la chanson n°", i, " :")

        Lire (T1[i])

    Jusqu'à (FN Verif (T1[i]))

    Fin pour

2) **Fin Chanson**

**TDOL**

| Nom   | Nature/Type | Rôle                                                                     |
|-------|-------------|--------------------------------------------------------------------------|
| i     | Entier      | Compteur                                                                 |
| Verif | Fonction    | Vérifier si une chaîne est composée uniquement de lettres et des espaces |

**b) Module Verif :**

0) **DEF FN Verif (ch : chaîne) : booléen**

1)  $j \leftarrow 0$

    Répéter

$j \leftarrow j + 1$

    Jusqu'à (NON (Majus (ch[j]) dans ["A".."Z", " "])) OU (j=Long(ch))

2)  $Verif \leftarrow$  (Majus (ch[j]) dans ["A".."Z", " "])

3) **Fin Verif**

**TDOL**

| Nom | Type   | Rôle     |
|-----|--------|----------|
| j   | Entier | Compteur |

**c) Module Saisir :**

0) **DEF PROC Saisir (Var N : entier)**

1) Répéter

    Ecrire (" Saisir le nombre de participants : ")

    Lire (N)

    Jusqu'à (N dans [5..100])

2) **Fin Saisir**

**d) Module Score :**

0) **DEF PROC Score (Var T2 : Tab2 ; N : entier )**

1) Pour i de 1 à 5 faire

$T2[i] \leftarrow 0$

    Fin pour

2) Pour i de 1 à N faire

    Répéter

        Ecrire (" Saisir le numéro de la chanson :")

        Lire (num)

    Jusqu'à (num dans [1..5])

$T2[num] \leftarrow T2[num] + 3$

    Fin pour

3) **Fin Score**

**TDOL**

| Nom | Nature/Type | Rôle                         |
|-----|-------------|------------------------------|
| i   | Entier      | Compteur                     |
| num | Entier      | Numéro de la chanson choisie |

e) **Module Trier :**

0) DEF PROC Trier (Var T2 : Tab2 ; Var T1 : Tab1 )

1) Pour i de 1 à 4 faire

Pmax ← FN Posmax (T2, i)  
 Si ( T2[Pmax] <> T2[i] )  
 Alors  
     Aux1 ← T2[Pmax]  
     T2[Pmax] ← T2[i]  
     T2[i] ← Aux1  
     Aux2 ← T1[Pmax]  
     T1[Pmax] ← T1[i]  
     T1[i] ← Aux2

Fin Si

Fin pour

2) Fin Trier

**TDOL**

| Nom    | Type     | Rôle                           |
|--------|----------|--------------------------------|
| i      | Entier   | Compteur                       |
| Pmax   | Entier   | Position maximum               |
| Aux1   | Entier   | Variable intermédiaire         |
| Aux2   | Chaine   | Variable intermédiaire         |
| Posmax | Fonction | Renvoie la position du maximum |

f) **Module Posmax :**

0) DEF FN Posmax (T2 : Tab2 ; i : entier)

1) Pm ← i

2) Pour j de i+1 à 5 faire

Si (T2[Pm] < T2[j])  
 Alors Pm ← j

Fin Si

Fin pour

3) Posmax ← Pm

4) Fin Posmax

**TDOL**

| Nom | Nature/Type | Rôle             |
|-----|-------------|------------------|
| j   | Entier      | Compteur         |
| Pm  | Entier      | Position maximum |

g) **Module Afficher :**

0) DEF PROC Afficher (T1 : Tab1; T2 : Tab2 )

1) Ecrire ("le classement est :")

Rang ← 1

Ecrire ("Rang ", Rang, " :", T1[1])

Pour i de 2 à 5 faire

Si (T2[i-1] = T2[i] )

Alors Ecrire (", ", T1[i])

Sinon

Rang ← Rang+1

Ecrire ("Rang ", Rang, " :", T1[i])

Fin Si

Fin pour

2) Fin Afficher

**TDOL**

| Nom  | Nature/Type | Rôle               |
|------|-------------|--------------------|
| i    | Entier      | Compteur           |
| Rang | Entier      | Rang de la chanson |