

Algorithmique II

Examen final

Durée : 1h 30 mn

Exercice 1 : (Sur 7 points)

Pour gérer le championnat marocain de football Botola Pro 1, nous créons :

- Le type Joueur, une structure regroupant les champs suivants :
 - NomJoueur, une chaîne de 30 caractères donnant le nom et le prénom du joueur,
 - DateNaissance, un tableau de 3 entiers donnant le jour, le mois et l'année de naissance du joueur,
 - Identité, une chaîne de 15 caractères donnant le CIN ou le numéro du Passeport du joueur,
 - ButsMarqués, un entier donnant le nombre de buts marqués par ce joueur durant la saison.
 - Le type Equipe comme une structure regroupant les champs suivants :
 - NomEquipe, une chaîne de 20 caractères donnant le nom de l'équipe,
 - Lesjoueurs, un tableau de 40 Joueur, donnant les joueurs de l'équipe,
 - NbPoints, un entier naturel donnant le nombre de points réalisés durant la saison,
 - NbMarqués, un entier donnant le nombre de buts marqués par l'équipe durant la saison,
 - NbEncaissés, un entier donnant le nombre de buts encaissés par l'équipe durant la saison.
 - BotolaPro1, un tableau donnant les 16 Equipe du championnat marocain Botola Pro1.
- A) Donner les déclarations de Joueur et Equipe comme nouveaux types de structure.
Et comme variable globale déclarer BotolaPro1 en tant que tableau d'Equipe.
- B) On suppose que toutes les données sont saisies dans le tableau BotolaPro1.
Ecrire une procédure Champion() qui détermine et affiche le champion du Botola Pro1.
[Indication : Le champion est celui qui a le nombre de points le plus grand. En cas d'égalité, c'est celui qui a le Goal-average le plus grand, c'est-à-dire qui a (NbMarqués-NbEncaissés) le plus grand]

Exercice 2 : (Sur 8 points)

On considère la procédure Partition donnée par :

Procédure Partition(T : Entier[1..N])

Var i, j : Entier

B : Boolleen

Début

i ← 1

Tant que (i < N) Faire

j ← i + 1

B ← Faux

Tant que (j ≤ N ET B=Faux) Faire

Si (T[j] = T[i]) Alors

B ← Vrai

Sinon

j ← j+1

Fin Si

Fin Tant que

```

    Si (B=Faux) Alors
        Ecrire("(", i, ",", i, ")", ", ")
        i ← i + 1
    Sinon
        Ecrire("(", i, ",", j, ")", ", ")
        i ← j + 1
    Fin Si
    Fin Tant que
Fin //Fin de la procédure

```

1. Donner les résultats d'application au tableau T= [8, 2, 6, 7, 1, 2, 6, 4] de la procédure Partition, ceci sous la forme suivante :

| i ^{ème} itération | j ^{ème} itération | i | T[i] | j | T[j] | B | Affichage |
|----------------------------|----------------------------|-----|------|-----|------|------|-----------|
| 1 ^{ère} itération | 1 ^{ère} itération | 1 | 8 | 2 | 2 | Faux | |
| | 2 ^{ème} itération | 1 | 8 | 3 | 6 | Faux | |
| | | ... | ... | ... | ... | ... | ... |
| ... | ... | ... | ... | ... | ... | ... | ... |

2. En déduire le but de la procédure Partition.
3. Déterminer l'ordre de grandeur de la complexité temporelle, dans les pires des cas, de la procédure Partition.

Exercice 3 : (Sur 5 points)

Soit A[1..N] un tableau d'entiers.

On considère la fonction réursive Inconnue (A : Entier[1..N], X : Entier, i : Entier) donnée par :

Fonction Inconnue(A : Entier[1..N], X : Entier, i : Entier) : Entier

Var m : Entier

Début

```

    Si (i = 1) Alors
        Retourner A[i]
    Sinon
        m = Inconnue(A, X, i-1)
        Si ( Abs(X-A[i]) < Abs(X-m) ) Alors
            Retourner A[i]
        Sinon
            Retourner m
    Fin Si
Fin Si

```

Fin

Où Abs est la fonction mathématique valeur absolue.

1. Calculer la valeur retournée par Inconnue(A, X, N), où N, A et X sont donnés par :
N=4, X=13 et A= [4, 9, 17, 5].
2. En déduire le but de la fonction Inconnue(A, X, i)