

<b>REPUBLIQUE TUNISIENNE</b> ◇◇◇ <b>MINISTERE DE L'EDUCATION</b>	<b>EXAMEN DU BACCALAUREAT</b> <b>SESSION DE JUIN 2012</b>		
	Epreuve : Algorithmique et Programmation	Durée : 3h	Coefficient : 2,25
<b>SECTION : Sciences de l'Informatique</b>		<b>SESSION DE CONTRÔLE</b>	

### Exercice 1 (3,25 points)

Soit **Temps** un type enregistrement, composé des trois champs suivants :

- **Heures** (avec  $0 \leq \text{Heures} \leq 23$ )
- **Minutes** (avec  $0 \leq \text{Minutes} \leq 59$ )
- **Secondes** (avec  $0 \leq \text{Secondes} \leq 59$ )

**Questions :**

- 1) Dresser le tableau de déclaration du type **Temps**.
- 2) Recopier le tableau ci-dessous sur votre feuille de copie et le compléter par les instructions algorithmiques correspondant aux traitements proposés, sachant que **T1** et **T2** sont deux variables de type **Temps**.

Traitements	Instructions algorithmiques
Saisir la variable <b>T1</b> , en respectant les contraintes nécessaires.	
Affecter le contenu de la variable <b>T1</b> à la variable <b>T2</b> .	
Afficher le contenu de la variable <b>T2</b> .	

### Exercice 2 (3 points)

Soient **x** et **a** deux réels donnés strictement positifs. On se propose de calculer la somme **S** définie par la formule suivante :

$$S = x - \frac{x^3}{a^2} + \frac{x^5}{a^4} - \frac{x^7}{a^6} + \frac{x^9}{a^8} - \dots$$

Ecrire un algorithme d'un module intitulé **CalcSom** permettant de calculer une valeur approchée de **S** à  $10^{-4}$  près.

### Exercice 3 (3 points)

Soit l'algorithme suivant, d'une procédure **TRI**, permettant de trier dans un ordre croissant un tableau **T** de **N** éléments, selon la méthode de tri par sélection :

```

0) DEF PROC TRI (Var T : tab ; N : entier)
1) Pour i de 1 à N-1 Faire
    pmin ← i
    Pour j de i+1 à N Faire
        Si (T[j] < T[pmin]) Alors
            pmin ← j
    Fin si
    Fin pour
    aux ← T[i]
    T[i] ← T[pmin]
    T[pmin] ← aux
  Fin pour
2) Fin TRI

```

Apporter les modifications nécessaires à l'algorithme de cette procédure pour que le tri soit **bidirectionnel**, c'est-à-dire qu'il permet à chaque fois de placer le plus grand élément vers la droite du tableau **T** et le plus petit élément vers la gauche.

#### Exercice 4 (4,5 points)

Ecrire une analyse d'un module intitulé **Symétrie** permettant de remplir une matrice **M** de  $(2*N-1)$  lignes et de **N** colonnes, comme le montre l'exemple ci-contre.

Sachant que :

- La partie inférieure de la matrice **M** (de la ligne **0** à la ligne **N-1**) représente les valeurs du **triangle de Pascal**.
- La partie supérieure de la matrice **M** (de la ligne **1-N** à la ligne **0**) représente la **symétrie** de la partie inférieure par rapport à la ligne **0**.

**Exemple :**

Pour **N = 5**

<b>M</b>	1	2	3	4	5
-4	1	4	6	4	1
-3	1	3	3	1	
-2	1	2	1		
-1	1	1			
0	1				
1	1	1			
2	1	2	1		
3	1	3	3	1	
4	1	4	6	4	1

#### Exercice 5 (6,25 points)

Etant donné un fichier texte intitulé "**MsgCrypte.txt**" contenant un texte crypté d'un message formé de **N** lignes, où chaque ligne représente un mot crypté.

Le principe du cryptage est le suivant :

- Chaque mot est crypté caractère par caractère.
- Chaque caractère est crypté par sa représentation binaire, sur 8 bits, de son code ASCII.

On se propose:

- de décrypter les lignes du fichier "**MsgCrypte.txt**" pour obtenir tous les mots formant le message initial,
- d'afficher le texte décrypté, en séparant les mots par des espaces.

**Exemple :**

Pour **N = 4** et le contenu du fichier "**MsgCrypte.txt**" suivant :

```
01000010 0110000101100011
0110100101101110011001101111
011010100 11101010110100101101110
00110010001100000011000100110010
```

Après décryptage, le programme affichera le message suivant : **Bac info juin 2012**

Notons que, le code ASCII du caractère "**B**" (le premier caractère du premier mot) est **66**, dont la représentation binaire est **01000010** (les huit premiers bits de la première ligne du fichier "**MsgCrypte.txt**").

**N.B** Le candidat n'est pas appelé à créer le fichier "**MsgCrypte.txt**".

**Travail à faire :**

1. Analyser le problème en le décomposant en modules.
2. Analyser chacun des modules envisagés.