

| | | |
|--|-----------------------|---|
| REPUBLIQUE TUNISIENNE MINISTRE DE L'EDUCATION ET DE LA FORMATION | SESSION PRINCIPALE | EXAMEN DU BACCALAURÉAT SESSION DE JUIN 2009 |
| SECTION : SCIENCES DE L'INFORMATIQUE | | |
| EPREUVE : ALGORITHMIQUE ET PROGRAMMATION | DURÉE : 3 h | COEF. : 2,25 |

Exercice 1 : (4 points)

On donne ci-dessous, l'algorithme de la procédure **TRI** dont le rôle est de trier par ordre croissant les éléments d'un tableau T de type W, vecteur de réels.

```

0) Procédure TRI (n : entier ; Var T : W)
1) Pour i de 2 à n Faire
    Si (T[i-1] > T[i]) Alors
        aux ← T[i]
        DECALER (i, j, T)
        T[j] ← aux
    FinSi
  FinPour
2) Fin TRI
    
```

DECALER est une procédure dont le rôle est de décaler d'un pas vers l'avant et à partir de l'indice i-1 les éléments de T jusqu'à trouver la position j où il faudra placer l'ancienne valeur T[i] sauvegardée dans aux.

- Comment appelle-t-on cette méthode de tri ?
- Rappeler brièvement le principe de cette méthode.
- Donner un algorithme de la procédure **DECALER** et préciser le type de transfert des paramètres utilisés.

Exercice 2 : (4 points)

Soit la fonction **inconnu1** suivante :

```

0- Fonction inconnu1 (x : réel ; n : entier) : réel
1- Si (n = 0) alors
    inconnu1 ← 1
  Sinon
    inconnu1 ← x * inconnu1(x, n-1)
  FinSi
2- Fin inconnu1
    
```

Soit la fonction **inconnu2** suivante :

```

0- Fonction inconnu2(n, p : entier) : entier
1- Si (p = 0) ou (p = n) alors
    inconnu2 ← 1
  Sinon
    inconnu2 ← inconnu2(n-1, p) + inconnu2(n-1, p-1)
  FinSi
2- Fin inconnu2
    
```

Questions :

- 1- Quel est l'ordre de récurrence de chacune des deux fonctions **inconnu1** et **inconnu2** ?
- 2- Exécuter manuellement **inconnu1**(2 , 3) et **inconnu2**(3 , 2).
- 3- Donner le rôle de chaque fonction.
- 4- Ecrire un algorithme d'une procédure permettant d'afficher à l'écran le développement du binôme $(x + 1)^n$ en un polynôme en x pour un entier n donné.
On rappelle que la formule du binôme est :

$$(x+1)^n = \sum_{k=0}^n C_n^k x^k = C_n^0 x^0 + C_n^1 x^1 + \dots + C_n^{n-1} x^{n-1} + C_n^n x^n$$

Sur l'écran l'affichage du terme x^k se fera sous la forme x^k .

Problème : (12 points)

Soit p et q deux entiers naturels tels que $5 \leq p \leq q \leq 32$. On se propose de remplir une matrice M à p lignes et q colonnes par des zéros (0) et des uns (1) de la façon suivante :

On mettra 1 dans toute cellule $M[i, j]$ si les représentations de i et de j dans la base 2 ont au moins une fois le chiffre 1 à la même position sinon on y mettra 0. On commencera à lire les positions à partir de la droite.

Exemples :

- $M[5, 3]$ aura la valeur 1 car les deux représentations binaires ont le chiffre 1 à la première position. En effet :

$$5 = (101)_2 \\ \text{et } 3 = (11)_2$$

- $M[9, 4]$ aura la valeur 0 car les deux représentations binaires n'ont pas de 1 à la même place. En effet :

$$9 = (1001)_2 \\ \text{et } 4 = (100)_2$$

On se propose ensuite de chercher tous les chemins rectilignes menant de la première ligne à la dernière ligne en n'empruntant que des cellules contenant des 1 et en sautant au maximum une seule cellule contenant 0. Par conséquent, un chemin rectiligne est une colonne ne comportant pas deux cellules consécutives contenant 0.

Pour tout chemin trouvé, on affichera son numéro (l'indice j). En plus, les données relatives à ce problème seront enregistrées dans un fichier texte intitulé **chemins.txt** et placé sur la racine du disque D. Ce fichier comportera dans sa première ligne le naturel p suivi d'une espace, suivie du naturel q et comportera dans chacune des lignes qui suivent le numéro du chemin trouvé. La dernière ligne contiendra le nombre de chemins trouvés.

On se propose d'écrire un programme qui réalise ces différents traitements.

Questions :

- 1) Analyser le problème en le décomposant en modules et déduire l'algorithme du programme principal qui permet de réaliser le traitement décrit précédemment.
- 2) Analyser chacun des modules envisagés précédemment et en déduire les algorithmes correspondants.