

# EXAMEN DU BACCALAUREAT - SESSION DE JUIN 2010

SECTION : Sciences de l'informatique

EPREUVE : Algorithmique et programmation DUREE : 3h COEFFICIENT : 2,25

Le sujet comporte 3 pages numérotées de 1/3 à 3/3

(Toutes les réponses se feront sur votre feuille de copie)

## Partie I : (10 points)

### Exercice 1 : (3 points)

Soit l'algorithme de la fonction **f** suivante :

0) Fonction **f** (**n**:entier ; **T**: TAB):.....

1) **i** ← 2

**f** ← vrai

   Répéter

      Si **T**[**i**-1] > **T**[**i**]

      Alors

**f** ← faux

      Sinon

**i** ← **i** + 1

      Finsi

   Jusqu'à (NON (**f**)) ou (**i** = **n** + 1)

2) Fin **f**

Avec **TAB** un type utilisateur représentant un tableau de 20 réels et **T** un tableau de **n** éléments

### Questions :

- 1) Donner le type de la fonction **f**
- 2) Soit le tableau **U** suivant :

U	2	3.5	.....	19.9	29
	1	2	3	4	5

et soit l'appel de la fonction **f** dans un programme :

Si **f**(**n**,**U**) Alors écrire ("les valeurs sont correctes")

   Sinon écrire ("les valeurs ne sont pas correctes")

FinSi

Proposer une valeur à mettre dans la 3<sup>ème</sup> case du tableau **U**, pour que le programme appelant affiche le message "les valeurs sont correctes".

- 3) Quel est le rôle de cette fonction ?

### Exercice 2 : (3 points)

Un entier **n** (**n** ≥ 10) est divisible par 25, si le nombre formé par ses deux derniers chiffres les plus à droites est égal à 00 ou 25 ou 50 ou 75.

**Exemple** : 679775 est divisible par 25 car le nombre formé par ces deux derniers chiffres est égal à 75.

Ecrire une analyse d'une fonction qui vérifie si un entier **n** est divisible par 25 en utilisant le principe décrit précédemment.

**Exercice 3 : (4 points)**

Ecrire l'algorithme d'un module permettant de calculer et d'afficher la valeur approchée de  $\pi$  à  $10^{-4}$  près en utilisant la formule suivante :

$$\pi = \frac{2}{1!} + \frac{(1!)^2 2^2}{3!} + \frac{(2!)^2 2^3}{5!} + \frac{(3!)^2 2^4}{7!} + \dots = \sum_{n=0}^{\infty} \frac{2^{n+1} (n!)^2}{(2n+1)!}$$

Avec  $n! = 1 * 2 * 3 * 4 * \dots * n$

Le calcul s'arrête lorsque la différence entre deux termes consécutifs devient inférieure ou égale à  $10^{-4}$

**Partie II : (10 points)**

On se propose d'écrire un programme permettant de :

- remplir une matrice carrée **M** de taille (**n x n**) selon le principe décrit ci-dessous. **n** est un entier donné ( $5 \leq n \leq 10$ )
- remplir, à partir de **M**, un fichier texte nommé **diagonal.txt** situé sur la racine du disque dur C:
- déterminer et afficher toutes les lignes de ce fichier qui contiennent **au moins quatre chiffres différents**

Le remplissage de la matrice **M** et du fichier **diagonal.txt** est décrit ci-dessous :

- 1) le remplissage de la matrice **M**, doit tenir compte des règles suivantes :
  - La première ligne de la matrice **M** est remplie, **d'une façon aléatoire (au hasard)**, par des chiffres de 1 à 9.
  - A partir de la deuxième ligne de la matrice **M**, un élément quelconque **M[l,c]** est déterminé en faisant la somme des éléments de la ligne (**l-1**), en commençant à partir de l'élément **M[l-1,c]**
  - Le nombre de cases remplies pour une ligne **l** est : **n - l + 1**.

	1	2	3	4	5	6
1	2	5	1	8	3	1
2	20	18	13	12	4	
3	67	47	29	16		
4	159	92	45			
5	296	137				
6	433					

**Exemple** : (voir la figure ci-contre)

$$M [2, 1] = 2 + 5 + 1 + 8 + 3 + 1 = 20$$

$$M [3, 4] = 12 + 4 = 16$$

- 2) Chaque ligne du fichier **diagonal.txt** contiendra les éléments de **M** se trouvant sur une diagonale droite, en commençant par celle à gauche et de haut en bas, de telle sorte que :
  - La ligne N°1 du fichier contient M [1, 1] c'est à dire **2**
  - La ligne N°2 du fichier contient M [1, 2] suivi de M [2, 1] c'est-à-dire **520**
  - La ligne N°3 du fichier contient M [1, 3] suivi de M [2, 2] suivi de M [3, 1] c'est-à-dire **11867**
  - .....

Pour l'exemple précédent, le contenu du fichier **diagonal.txt** sera comme suit :

```
2
520
11867
81347159
3122992296
141645137433
```

et les lignes qui seront affichées sont :

```
11867
81347159
3122992296
141645137433
```

En effet :

- La ligne numéro 1 n'est pas affichée car elle ne contient qu'un seul chiffre.
- La ligne numéro 2 n'est pas affichée car on ne trouve que 3 chiffres distincts.
- Les lignes numéro 3, 4, 5 et 6 seront affichées car elles contiennent chacune au moins 4 chiffres distincts.

### **Travail demandé**

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.