Série de TP n°6

Exercice 1

Ecrire un programme qui

a) saisit un entier `n` (0<`n`<20, faire une fonction qui effectue une telle saisie),

b) saisit, dans un tableau, `n` nombres entiers,

c) puis qui successivement calcule :

- la moyenne des éléments du tableau (faire une fonction qui retourne la moyenne d'un tableau d'entiers de `n` éléments), puis affiche la moyenne et les éléments du tableau, avec leur rang, qui sont supérieurs à la moyenne,
- la valeur minimum du tableau (faire une fonction qui retourne ce minimum) puis affiche le minimum et tous les rangs dans le tableau où il est atteint.
- le nombre maximum de valeurs consécutives du tableau qui sont strictement positives.

Exercice 2

Un polynôme à coefficients réels

$$P(x) = a_0 + a_1x^1 + a_2x^2 + \cdots + a_{n-1}x^{n-1} + a_nx^n$$

peut être représenté sous la forme d'un tableau dont chaque élément correspond à un des monômes du polynôme P. Le but de cet exercice est d'écrire un programme permettant, au choix de l'utilisateur, de réaliser une des manipulations de base suivantes sur des polynômes :

a) saisie de deux polynômes, calcul et affichage de l'addition de ces deux polynômes,

b) saisie d'un polynôme, calcul et affichage de sa dérivée,

c) saisie d'un polynôme, d'un réel α puis affichage de l'évaluation du polynôme en α par l'algorithme de Hörner présenté ci-dessous.

Pour cela, on écrira les fonctions permettant :

a) la saisie au clavier d'un polynôme dont on connaît le degré,

b) l'affichage d'un polynôme à l'écran,

c) l'addition de deux polynômes,

d) le calcul de la dérivée d'un polynôme,

e) l'évaluation d'un polynôme en un point $\alpha \in \mathbb{R}$ par l'algorithme de Hörner.

L'algorithme de Hörner permet d'évaluer rapidement un polynôme en un point $\alpha \in \mathbb{R}$. Il est basé sur l'écriture suivante du polynôme P

$$P(x) = a_0 + x(a_1 + x(a_2 + \cdots + x(a_{n-2} + x(a_{n-1} + xa_n))\cdots)).$$

On remarquera dans cette écriture l'imbrication de termes de la forme :

$$a_i + x{*}(\textit{un terme de la même forme}).$$

Exercice 3

Ecrire un programme

1. demandant à l'utilisateur l'ordre $n \in \{2, 3, \ldots, 10\}$ d'une matrice et la forme de celle-ci qu'il souhaite parmi (i et j désignent des entiers quelconques entre 1 et n et les exemples correspondent au cas $n = 3$) :

forme	contraintes	exemple
scalaire :	$i \neq j \Longrightarrow a_{i,j} = 0, \text{ et } a_{i,i} = a_{j,j}$	$\begin{pmatrix} 5 & 0 & 0 \\ 0 & 5 & 0 \\ 0 & 0 & 5 \end{pmatrix}$
diagonale :	$i \neq j \Longrightarrow a_{i,j} = 0$	$\begin{pmatrix} 1 & 0 & 0 \\ 0 & -8 & 0 \\ 0 & 0 & 3 \end{pmatrix}$
diagonale 2 :	à vous de l'écrire...	$\begin{pmatrix} 0 & 0 & 7 \\ 0 & -2 & 0 \\ 9 & 0 & 0 \end{pmatrix}$
triangulaire supérieure :	$i > j \Longrightarrow a_{i,j} = 0$	$\begin{pmatrix} 4 & 0 & -1 \\ 0 & 1 & 2 \\ 0 & 0 & -7 \end{pmatrix}$
triangulaire inférieure :	à vous de l'écrire...	$\begin{pmatrix} -3 & 0 & 0 \\ 5 & 4 & 0 \\ 7 & -5 & 1 \end{pmatrix}$
quelconque :	aucune	$\begin{pmatrix} 6 & 8 & -9 \\ 0 & -5 & 3 \\ -4 & 2 & 7 \end{pmatrix}$

2. génère aléatoirement les coefficients (entiers entre -9 et 9) d'une matrice de cette forme et affiche cette matrice.

Exercice 4

Le but du programme à écrire est de gérer une liste d'au plus 50 points de l'espace (un point est formé de 3 réels).

Ecrire le programme qui permet la saisie d'une liste de points par l'utilisateur, la saisie s'achèvera lorsque l'utilisateur rentrera le point (0,0,0), puis qui, au choix de l'utilisateur (l'utilisateur pourra effectuer un ou plusieurs traitements) :

a) affiche les points sous le format :

	point 1	point 2	$\cdots$	point P
x	12.360	6.450	$\cdots$	6.200
y	-4.850	89.470	$\cdots$	0.100
z	5.400	-47.580	$\cdots$	9.100

b) retire, si la liste n'est pas vide, un point de la liste par la saisie de son numéro,

c) ajoute, si la liste n'est pas pleine, un nouveau point saisi par l'utilisateur,

d) calcule la distance entre 2 points i et j de la liste donnés par l'utilisateur (écrire une fonction), puis affiche cette distance.

<u>Rappel</u> : distance(point i, point j)=$\sqrt{|x_i - x_j|^2 + |y_i - y_j|^2 + |z_i - z_j|^2}$