

TRAVAUX DIRIGÉS - 2

1 Les tableaux statiques

EXERCICE 1

Listing 1: exo1ex.c

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <time.h>
4
5 int main(){
6     int i;
7     time_t t1;
8     /******
9     /* Initialisation du generateur de V.A. */
10    time(&t1);
11    srand48((long) t1);
12    /******
13
14    /* Affichage de 5 nbs aleatoires uniformement distribues sur [0,1] */
15    for (i=0;i<5;i++)
16        printf("%lf\n", drand48 ());
17    return 1;
18 }
```

- Q. 1** Donner la(les) commande(s) permettant de compiler le programme `exo1ex.c`. ■
- Q. 2**
1. Ecrire un programme permettant de générer un tableau de N nombres aléatoires uniformément distribués sur l'intervalle $[0, 1]$ et d'afficher leur moyenne. Le nombre N est saisi au clavier. ■
 2. A l'aide de diagrammes de représentation mémoire, expliquer le déroulement de ce programme. ■
- Q. 3**
1. Ecrire un programme structuré utilisant des fonctions réalisant les mêmes opérations. ■
 2. A l'aide de diagrammes de représentation mémoire, expliquer le déroulement de ce programme. ■

EXERCICE 2

Soient $\mathbf{u} = \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}$ et $\mathbf{v} = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$ deux vecteurs.

- Q. 1** En stockant les différents vecteurs dans des tableaux, écrire un programme permettant de
- calculer le vecteur $\mathbf{w} = \mathbf{u} + \mathbf{v}$,
 - afficher \mathbf{w} ,
 - calculer le produit scalaire de \mathbf{u} par \mathbf{v} ,
 - l'afficher. ■

Soient $\mathbb{A} = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix}$ une matrice et $\mathbf{x} = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$ un vecteur.

Q. 2 En utilisant un tableau bidimensionnel pour stocker la matrice \mathbb{A} et un tableau monodimensionnel pour stocker le vecteur \mathbf{x} , écrire un programme permettant de calculer le vecteur $\mathbf{y} = \mathbb{A}\mathbf{x}$ puis l'afficher. ■

2 Les structures

EXERCICE 3

Q. 1 Définir le type `Complexe` correspondant à une structure de champ `Re` et `Im`, respectivement partie réelle et partie imaginaire. ■

Soient $a = 1 - i$ et $b = -2 + 3i$ deux nombres complexes.

Q. 2 Ecrire un programme permettant de

- calculer le complexe $c = a + b$,
- afficher c ,
- calculer le produit $d = a * b$,
- afficher d . ■

Soient $z_k = 2k + i(k - 1)$, $\forall k \in \llbracket 1, 20 \rrbracket$.

Q. 3 Ecrire un programme permettant de

- créer un tableau de complexes contenant les z_k .
- calculer $S = \sum_{k=1}^{20} z_k$,
- afficher S . ■

EXERCICE 4

Soit $\mathbf{x} \in \mathbb{R}^n$ le vecteur défini par

$$x_i = \cos\left(\frac{i\pi}{2n}\right), \forall i \in \llbracket 1, n \rrbracket.$$

Q. 1 Définir un type `Vector` correspondant à une structure de champs :

- `dim` : dimension du vecteur,
- `val` : tableau contenant les composantes du vecteur ■

Q. 2 1. Ecrire un programme permettant de

- saisir au clavier la valeur de n ,
- créer le vecteur \mathbf{x} ,
- afficher le vecteur \mathbf{x} ,
- calculer $S = \sum_{i=1}^n x_i$,
- afficher S .

2. A l'aide de diagrammes de représentation mémoire, expliquer le déroulement de ce programme. ■

```

1 #include <stdio.h>
2 #include <math.h>
3 #define DIMMAX 1000
4 typedef struct vecteur{
5     int dim;
6     double val[DIMMAX];
7 } Vector;
8 void PrintVector(Vector V, const char Name[]);
9 double SommeVector(Vector V);
10 void InitVector(Vector *pV);
11 void InitVector0(Vector V);
12 Vector InitVector1(void);
13
14 int main(){
15     Vector X;
16     int i;
17     double S;
18
19     InitVector0(X);
20     PrintVector(X, "X");
21     InitVector(&X);
22     PrintVector(X, "X");
23     X=InitVector1();
24     PrintVector(X, "X");
25
26     /* Affichage de X */
27     PrintVector(X, "X");
28
29     /* Calcul de S */
30     S=SommeVector(X);
31
32     printf("S=%lf\n",S);
33
34     return 1;
35 }
36
37 void InitVector(Vector *pV){
38     int i;
39     printf("n="); scanf("%d",&pV->dim);
40     /* Initialisation du vecteur X */
41
42     for (i=0;i<pV->dim;i++)
43         pV->val[i]=cos(i*M_PI/(2*pV->dim));
44 }
45 void InitVector0(Vector V){
46     int i;
47     printf("n="); scanf("%d",&V.dim);
48     /* Initialisation du vecteur X */
49
50     for (i=0;i<V.dim;i++)
51         V.val[i]=i+1.;
52 }
53
54 Vector InitVector1(void){
55     int i;
56     Vector V;
57     printf("n="); scanf("%d",&V.dim);
58     /* Initialisation du vecteur X */
59
60     for (i=0;i<V.dim;i++)
61         V.val[i]=V.dim-i;
62     return V;
63 }
64
65 void PrintVector(Vector V, const char3 Name[]){
66     int i;
67     printf("dim=%d\n",V.dim);
68     for (i=0;i<(V.dim>10?10:V.dim);i++) /* Affichage limite au 10 premieres valeurs*/
69         printf("%s [%d]=%lf\n",Name,i,V.val[i]);
70 }

```

Q. 3 *A l'aide de diagrammes de représentation mémoire, expliquer le déroulement de ce programme.* ■