

LANGAGE C AVANCÉ POUR LA SIMULATION NUMÉRIQUE

EXAMEN DU 10 JANVIER 2013

Durée : 3h

**Aucun document autorisé
Aucun appareil électronique autorisé**

EXERCICE 1 : (4 points)

Listing 1: exo9.c

```
1 #include <stdio.h>
2 #include <math.h>
3
4 void f1(int i, int *j, double *x, double y);
5
6 int main(){
7     int i=1,j=2;
8     double x=3.,y=M_PI;
9
10    f1(i,&j,&x,y);
11    printf("main: i=%6d, j=%6d, x=%.16lf, y=%.16lf\n", i, j, x, y);
12    f1(j,&i,&y,x);
13    printf("main: i=%6d, j=%6d, x=%.16lf, y=%.16lf\n", i, j, x, y);
14    return 1;
15 }
16
17 void f1(int i, int *j, double *x, double y){
18     (*j)++;
19     i--;
20     *x=y/2.;
21     y*=2;
22     printf("f1: i=%6d, j=%6d, x=%.16lf, y=%.16lf\n", i, *j, *x, y);
23 }
```

Q. 1 A l'aide de diagrammes de représentation mémoire, expliquer le déroulement du programme et donner les résultats affichés. ■

Q. 2 Donner les commandes Linux/Unix permettant d'exécuter le code précédent. ■

EXERCICE 2 : (8 points)

Q. 1 Ecrire la fonction *GetData* d'entrée des données a , b et n avec $a \in \mathbb{R}$, $b \in \mathbb{R}$ et $n \in \mathbb{N}^*$. Elles seront saisies au clavier par l'utilisateur. ■

Q. 2 Ecrire une fonction *InitMesh1D* permettant d'initialiser $h = (b-a)/n$ et un tableau alloué dynamiquement contenant l'ensemble des $x_i = a + ih$, $i \in \{0, \dots, n\}$. ■

Q. 3 Ecrire une fonction *CalculF* permettant d'initialiser un tableau alloué dynamiquement contenant l'ensemble des valeurs $f(x_i)$, $i \in \{0, \dots, n\}$ où f est une fonction réelle à valeurs réelles supposée donnée (paramètre de la fonction *CalculF*). ■

Q. 4 *Ecrire un programme utilisant les fonctions précédentes, sachant que la fonction f de `CalculF` sera $f(x) = \cos(2x)$, et que l'on affichera l'ensemble des couples $(x_i, f(x_i))$ (un couple par ligne).* ■

On va maintenant reprendre les questions précédentes en utilisant la structure `mesh1D` de champs :

N : nombre de discrétisation n ,

X : tableau dynamique contenant l'ensemble des x_i ,

H : pas de discrétisation h ,

Q. 5 *Donner la déclaration de la structure `mesh1D` et définir le type `MESH1D` correspondant à la structure `mesh1D`.* ■

Q. 6 *Ecrire une fonction `InitMESH1D` permettant d'initialiser une variable de type `MESH1D` à partir des données obtenues par la fonction `GetData`.* ■

Q. 7 *Ecrire une fonction `CALCULF` permettant d'initialiser un tableau alloué dynamiquement contenant l'ensemble des valeurs $f(x_i)$, $i \in \{0, \dots, n\}$ où f est une fonction réelle à valeurs réelles supposée donnée (paramètre de la fonction `CALCULF`) et en utilisant une variable de type `MESH1D` supposée donnée.* ■

On souhaite maintenant calculer une approximation de l'intégrale $\int_a^b f(x)dx$ en utilisant la formule composite des trapèzes :

$$\int_a^b f(x)dx \approx I_n(f) = h \left(\frac{f(x_0) + f(x_n)}{2} + \sum_{i=1}^{n-1} f(x_i) \right).$$

Q. 8 *Ecrire une fonction `TRAPEZE` permettant de calculer l'approximation de l'intégrale par la méthode précédente.* ■

Q. 9 *Ecrire un programme utilisant les fonctions précédentes, sachant que la fonction f de `CALCULF` sera $f(x) = \sin(3x)$, et que l'on affichera l'ensemble des couples $(x_i, f(x_i))$ (un couple par ligne) ainsi que l'approximation de l'intégrale donnée par la fonction `TRAPEZE`.* ■

EXERCICE 3 : (8 points)

Dans cet exercice, nous allons implémenter et utiliser quelques outils d'algèbre linéaire.

Q. 1 *Définir le type `vector` correspondant à une structure ayant pour champs :*

- `dim` : dimension du vecteur (type `int`),
- `pval` : pointeur sur les composantes du vecteur (type `double *`). ■

Q. 2 *Nous voulons écrire une fonction `AllocVector` permettant d'allouer un objet de type `vector` à la dimension N .*

1. Donner deux prototypes possibles (non trivialement similaire) pour la fonction `AllocVector`.
2. Choisir un prototype et écrire la fonction associée. ■

Q. 3 1. *Ecrire une fonction `freeVector` permettant de désallouer le champ `val` d'un objet de type `vector`.*

2. Justifier le choix de ses arguments! ■

Q. 4 1. *Ecrire une fonction `FREEVector` permettant de désallouer le champ `val` d'un objet de type `vector`, de mettre à zéro le champ `dim` et à `NULL` le champ `val`.*

2. Justifier le choix de ses arguments! ■

Q. 5 Nous voulons écrire une fonction `RandVector` permettant d'initialiser chacune des composantes d'un vecteur par un nombre aléatoire (utiliser la fonction `drand48()` du langage C qui retourne un `double` entre 0 et 1 : `double drand48 (void);`). Le vecteur est supposé déjà alloué.

1. Donner deux prototypes possibles (non trivialement similaire) pour la fonction `RandVector`.

2. Choisir un prototype et écrire la fonction associée. ■

Q. 6 Soient $\mathbf{u} = \begin{pmatrix} u_1 \\ \vdots \\ u_n \end{pmatrix} \in \mathbb{R}^n$, $\mathbf{v} = \begin{pmatrix} v_1 \\ \vdots \\ v_n \end{pmatrix} \in \mathbb{R}^n$ et α, β deux réels. Les fonctions `ScalarProd`, `aUpbV` et `Norm` correspondent à :

$$\text{ScalarProd} : \langle \mathbf{u}, \mathbf{v} \rangle = \sum_{i=1}^n u_i v_i,$$

$$\text{aUpbV} : \alpha \mathbf{u} + \beta \mathbf{v},$$

$$\text{Norm} : \|\mathbf{u}\| = \left(\sum_{i=1}^n u_i^2 \right)^{1/2}.$$

1. Pour chacune des trois fonctions, donner deux prototypes possibles (non trivialement similaire).

2. Pour chacune des trois fonctions, choisir un prototype et écrire la fonction associée. ■

Q. 7 Écrire un `main` simple et fonctionnel utilisant au moins les fonctions `RandVector`, `ScalarProd`, `aUpbV` et `Norm`. ■