MASTER 2 MPM **Université Paris XIII**
Année 2012-2013 **Institut Galilée**

Langage C avancé pour la simulation numérique

Examen du 5 avril 2013

Durée : 3h

Aucun document autorisé
Aucun appareil electronique autorisé

Exercice 1 : (4 points)

Listing 1: exo12.c

```
1  #include <stdio.h>
2  #include <math.h>
3
4  void f(int i, int *j);
5  void g(double *x, double y);
6
7  int main(){
8    int i=1,j=2;
9    double x=1.,y=sin(M_PI);
10   printf("1) main: i=%6d, j=%6d, x=%.16lf, y=%.16lf\n",i,j,x,y);
11   f(i,&j);g(&x,y);
12   printf("2) main: i=%6d, j=%6d, x=%.16lf, y=%.16lf\n",i,j,x,y);
13   f(j,&i);g(&y,x);
14   printf("3) main: i=%6d, j=%6d, x=%.16lf, y=%.16lf\n",i,j,x,y);
15   return 1;
16 }
17
18 void f(int i, int *j){
19   (*j)++;
20   i--;
21   printf("  -> f : i=%6d, *j=%6d\n",i,*j);
22 }
23
24 void g(double *x, double y){
25   *x=3*y-*x;
26   y*=2;
27   printf("  -> g : *x=%.16lf, y=%.16lf\n",*x,y);
28 }
```

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

Q. 2 *Donner les commandes Linux/Unix permettant de compiler et d'exécuter le code précédent.* ▪

Exercice 2 : (2 points)

Ecrire un programme permettant successivement de :

1. demander à l'utilisateur de rentrer successivement deux entiers qui seront stockés dans les variables i et j,
2. afficher les valeurs de ces deux variables,
3. permutter le contenu des deux variables à l'aide d'une fonction `permut`,

4. afficher les valeurs des deux variables i et j.

Voici un exemple de l'affichage obtenu :

```
i=2
j=3
Before permut function : i=2 and j=3
After  permut function : i=3 and j=2
```

Exercice 3 : (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 $x_i = a + ih,\ i \in \{0, \ldots, n\}$.* ▪

Q. 3 *Ecrire une fonction* `CalculF` *permettant d'initialiser un tableau alloué dynamiquement contenant l'ensemble des valeurs $f(x_i),\ i \in \{0, \ldots, n\}$ 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 $f(x_i),\ i \in \{0, \ldots, n\}$ 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 donnée par $I_n(f)$.* ▪

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 4 : (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 une variable 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'une variable de type* `vector`*.*

2. *Justifier le choix de ses arguments!* ▪

Q. 4 1. *Ecrire une fonction* `FREEVector` *permettant de désallouer le champ* `val` *d'une variable 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* `SequenceVector` *de paramètres a et b, deux réels $a < b$ (entre autes). Cette fonction initialise une variable de type* `vector` *pour quelle corresponde au vecteur $\boldsymbol{x} \in \mathbb{R}^N$ défini pour tout $i \in [\![1, N]\!]$ par $\boldsymbol{x}_i = a + (i-1) * h$ avec $h = (b-a)/(N-1)$.*

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

Q. 6 *Soient* $\boldsymbol{X} = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} \in \mathbb{R}^n$, $\boldsymbol{Y} = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix} \in \mathbb{R}^n$ *et a un réel. Les fonctions* `ScalarProd`*,* `aXpY` *et* `NormInf` *correspondent à :*

`ScalarProd`	:	$\langle \boldsymbol{X}, \boldsymbol{Y} \rangle = \sum_{i=1}^{n} X_i Y_i,$
`aXpY`	:	$a\boldsymbol{X} + \boldsymbol{Y},$
`NormInf`	:	$\|\boldsymbol{u}\|_\infty = \max_{i \in [\![1,N]\!]} \|X_i\|.$

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 *Ecrire un* `main` *simple et fonctionnel utilisant au moins les fonctions* `SequenceVector`*,* `ScalarProd`*,* `aXpY` *et* `NormInf`*.* ▪