INSA - 3GE **ANNEE 2014/ 2015**

Module IF1

INFORMATIQUE INDUSTRIELLE

Durée 3 heures

(environ 2h pour les Systèmes Logiques et environ 1h pour le C)

Documents papiers autorisés

A rédiger sur des copies séparées

1 copie Systèmes Logiques
1 copie C

Langage C

I- Questions pièges (vous êtes prévenus !), répondre en une ligne

1) Quelle est la taille maximale d'un tableau de char ?
2) Parmi les 4 variables a, b, c et d suivantes quelle est celle occupant la plus grande place mémoire ? char *a ; short *b ; int *c ; double *d ;
3) Si on a : char t[]=''hello'' ; que sera affiché par l'instruction cout << (int) t[5] ; ?

II- Exercice passage de paramètre à des fonctions

a) Soit la fonction *Calcul* suivante:

```
void Calcul( short taille, double ***R, float **M, int *v);
```

À partir des variables déclarées dans le main suivant, compléter l'appel à la fonction *Calcul* :

```
int main()
{
short N = 5;
int *vector_int = new int[N];
char choix;
float coef[5][]={{1,2,3,4,5},{5,1,2,3,4},{4,5,1,2,3},{3,4,5,1,2},{2,3,4,5,1}};
double **matrice;
for(int i=0; i<N; i++)
  {     cout << " i = "; cin >> vector_int[i];        }

Calcul(      ) ; //<-- à compléter

… //suite du code du main
```

b) Modification de valeur de pointeur

Ecrire la fonction Swap_Ad qui change ce sur quoi pointe le pointeur pt en l'affectant à l'adresse val_ad. On donne son utilisation dans un main :

```
        double pi = 3.14159265 ;
        double e = 2.718281;
        double *pt = & pi ;
        Swap_Ad(&pt, &e) ;
```

III- Exercice I : affichage sans répétition de valeurs

Ecrire la fonction ***Affiche*** qui permet d'afficher le contenu d'un tableau d'entiers où chaque valeur est séparée par un espace et en remplaçant les répétitions consécutives d'une valeur par le caractère « . ».

Exemple : pour les valeurs 1 2 2 2 3 3 2 4 4 5 5 5

On a l'affichage suivant: 1 2 . . 3 . 2 4 . 5 . .

IV- Analyse de code

Soit le programme suivant:

```
#include <iostream>
#include <cmath>
using namespace std;

void SumOfTerms(int *H, int sum, int S, int N,int OriginalS)
{
    if (S == 0 )
        H[sum-OriginalS]++;
    else
        {
        S--;
        for(int i=1; i<N+1; i++)
            SumOfTerms(H,sum+i,S,N,OriginalS);
        }
}

int main(int argc, char *argv[])
{
    int S; // number of sum   : x + x + ...
    cout << "Valeur de S = ";
    cin >> S ;

    int N; // sequence 1..N
    cout << "Valeur de N = ";
    cin >> N ;

    int Size_H = N*S-S+1;
    int *H = new int[Size_H];
    int i;
// Initialisation
    for(i=0; i<Size_H; i++)
         H[i]=0;

// Calcul
    for(i=1;i<N+1;i++)
         SumOfTerms(H,i,S-1,N,S);

// Affichage
    for(i=0; i<Size_H; i++)
         cout<< H[i] << " ";
    cout << endl;

    delete[] H;
    return 0;
}
```

1) Quel est le rôle de ligne int *H = new int[Size_H]; ?

2) Donner l'affichage produit à l'exécution de ce programme si l'utilisateur entre 2 et 3 (S=2 et N=3).

3) Avec N et S quelconques, exprimer le nombre d'exécutions de la ligne : H[sum-OriginalS]++; ?

4) L'exécution de la ligne H[sum-OriginalS]++; prend 1ns sur un pc. Combien de temps, au minimum, va devoir attendre l'étudiant du premier cycle exécutant ce programme avec N=10 et S=100 ?

Exercice II : Opérateur *vec*

En mathématique, la vectorisation d'une matrice **A** de taille *m* x *n* permet d'écrire cette matrice sous forme d'un vecteur colonne de taille *m.n* x 1. Exemple, attention aux indices :

$$\mathbf{A} = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}, \quad vec(\mathbf{A}) = \begin{bmatrix} a_{11} & a_{21} & a_{12} & a_{22} \end{bmatrix}^T$$

On donne la déclaration de la fonction ***vec***

```
double * vec(const double **A, int m, int n) ;
```

Remarques :

- Dans la déclaration précédente, **A** est une matrice, on accède à l'élément $a_{i,j}$ par A[i][j].
- La fonction ***vec*** doit retourner un tableau alloué dynamiquement dans la fonction.

1) A partir de la déclaration de la fonction ***vec*** et des remarques, écrire sa définition (le code de la fonction).
2) Ecrire une fonction permettant d'afficher un vecteur obtenu par la fonction ***vec***.
3) Dans une fonction ***main***, illustrer l'utilisation de la fonction ***vec*** avec une matrice 3x2.
4) Penser à la libération de la mémoire (dé-allocation) dans votre ***main***…

Exercice III : Fichiers

On dispose d'un fichier texte « *note.txt* » comportant le prénom et le nom d'un étudiant sur une ligne, puis, sur la ligne suivante, une série d'exactement 5 notes.

Exemple de fichier *note.txt* :

```
Yves Bizangouin
12 14.5 10 8 15.5
```

Ecrire le programme permettant de lire ce fichier et d'afficher le nom et le prénom de l'étudiant ainsi que la moyenne calculée sur les 5 notes.

Exemple d'exécution :

```
Bizangouin Yves : 12
```