Université de Lorraine NOM et Prénom: .....

Examen Programmation Avancée

Supports de cours autorisés, appareils électroniques interdits.
Durée : 2h

Exercice 1

Q1) Déclarez de manière dynamique un tableau à deux dimensions de taille 10x10 et puis l'initialiser tel que l'élément situé à la $i^{ème}$ ligne et la $j^{ème}$ colonne contient l'entier `i*j`.

Q2) Donnez la dé-allocation du tableau à deux dimensions dans la Q1.

Q3) Pour un tableau à une dimension, donnez deux manières d'accéder à l'élément `i` de ce tableau.

Q4) Dites ce qu'affiche le code suivant :

```
int i = 12;
printf("i=%d\n", ++i);
int j = 9;
printf("j=%d\n", j++);
```

Q5) Dites ce qu'affiche le code suivant :

```
#include <stdio.h>
int f(int x) {
  x *= 2;
  return x;
}
int g(int *x) {
  *x += 5;
  return *x;
}
void main() {
  int a = 10;
  int b = f(a);
  int c = g(&b);
  int d = f(g(&a));
  int e = f(f(g(&c)));
  printf("a=%d, b=%d, c=%d, d=%d, e=%d\n", a, b, c, d, e);
}
```

Q6) Dites ce que fait le code suivant :

```
#include <stdio.h>
int f(int m){
   if(m<=0)
        return 1;
  return m*f(m-1);
}
void main(){
   int m;
   scanf("%d", &m);
  printf("f(%d)=%d", m, f(m));
}
```

Q7) Que contiennent les variables `u` et `v` à la fin du programme ?

```
int u = 5,v = 7;
u = u + v ;
v = u – v ;
u = u – v ;
```

Q8) Proposez un code récursif permettant de calculer le n^{ème} terme, F_n, de la suite de Fibonacci. Rappel : la suite de Fibonacci est définie par :

$$\begin{cases} F_0 = 0 \\ F_1 = 1 \\ F_n = F_{n-2} + F_{n-1} \end{cases}$$

Q9) Dites ce qu'affiche le code suivant :

```
#include <stdio.h>
void main() {
  int *x, *y;
  int z = 3;
  x = &z;
  y = x;
  z += 3;
  *x *= *y + 3;
  printf("x=%d, y=%d, z=%d \n", *x, *y, z);
}
```

Q10) Soit n un entier. Proposez un code C renvoyant le nombre de bits requis pour une représentation binaire de n.

Exercice 2

Dans cet exercice on s'intéresse à la structure de graphe non orienté et pondéré. Plus précisément, le graphe G = (V, E) est défini par un ensemble de sommets V et un ensemble d'arêtes E, tel que chaque arête e ∈ E connecte deux sommets v_1, v_2 ∈ V et est associée à une valeur positive, appelée le poids de l'arête e. La Figure 1 montre un exemple du graphe non orienté et pondéré.

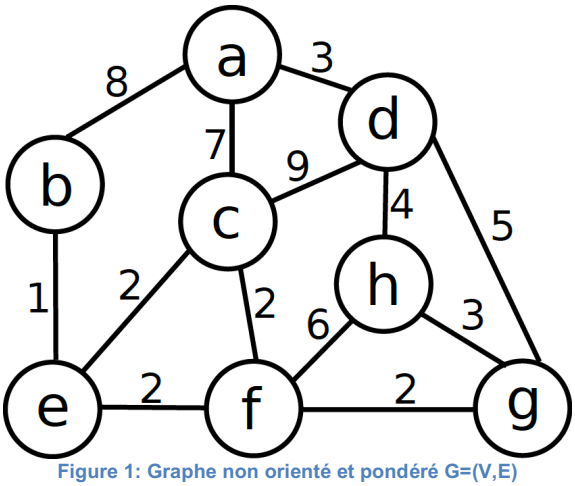

Figure 1: Graphe non orienté et pondéré G=(V,E)

Q1) Donnez les trois représentations du graphe G dans la Figure 1 avec la matrice d'adjacence, la matrice d'incidence et la liste d'adjacence. Expliquez dans quel cas, les représentations avec les matrices sont plus avantageuses que la liste d'adjacent.

Q2) Considérons un graphe non-orienté et pondéré G d'ordre n (c'est-à-dire G contient n sommets). Proposez une structure de données en C/C++ (avec ***struct*** et ***typedef***) pour stocker le graphe G avec la représentation par la liste d'adjacence.

Q3) Même question pour la représentation de G avec la matrice d'adjacence.

Q4) Que signifie l'arbre de recouvrement minimal (ARM) d'un graphe non orienté et pondéré ? Quel est l'ARM du graphe G dans la Figure 1 ? Proposez une implémentation en C/C++ l'algorithme de Krusal pour trouver l'ARM d'un graphe non-orienté et pondéré G.