Ecole préparatoire en Sciences et Techniques d'Oran
Intitulé : Algorithmique 2 et structures de données avancées — Sigle : Algo2
2^{me} année, Semestre 3, 2011/2012 — Devoir surveillé
Décembre 2011 — Durée : 02h00mn

Partie cours : 10pts

Exercice 1 : (4.5pts)

1. Soit le programme suivant :

```
int x=3;
int * px;
int y=-10;
int * py=&y;
px = &x;
*py = y + *px;
x = *py-x;
cout << "x=" << x << endl;
```

   Q : Quelle est la valeur de x ?

2. Donnez la syntaxe générale de l'allocation dynamique en C++.
3. Soit le programme suivant :

```
int tab[10]={5,8,4,3,9,6,5,4,3,8};
printf("%d \n",*tab);
printf("%d \n",tab[0]);
int *pt = tab;
pt++;
printf("%d \n",pt[0]);
```

   Q : que va afficher les trois instructions d'affichage ?

4. Soit le prototype de la fonction g :

   ```
   void g(int j, bool b=true, float y=1.1);
   ```

   Q : donnez les différents cas d'appels corrects de cette fonction ?

5. Que permet de générer la compilation séparée en programmation modulaire ?

Exercice 2 : 5.5pts

Soit le programme suivant :

```
#include <stdlib.h>
#include <iostream>
using namespace std;

void h(int a, int b);
main ( )
{
    int i(6),j(-23);
    h(i,j);
    cout << "après application de h : i=" << i << " et j=" << j << endl;
}
void h(int a, int b)
{
    if (a>b)
    {
        a=a+b;
        b=a-b;
        a=a-b;
    }
}
```

1. Que fait la fonction h ?
2. Quels sont les contenus de i et j, après l'appel de la fonction h ? Pourquoi ?
3. Si le programme ci-dessus présente des erreurs, quelles sont les solutions possibles? Donner le programme résultant pour chaque solution ?

Partie TD : 10pts

Exercice 1 : 2.5 pts

Soit la partie suivante d'un programme C++ :

```
const int N=20;
typedef bool MC[N][N];
bool fct(MC T, int taille = N)
{
    for(int i=0; i<taille-1;i++)
    {
        for(int j=i+1; j<taille; j++)
        {
            if (i != j)
            {
                int k=0;
                while (T[i][k] == T[j][k])
                {
                    if (k == taille-1) return true;
                    else k = k+1;
                }
            }
        }
    }
    return false;
}
```

Que fait la fonction *fct* ? justifiez.

Exercice 2 : 5 pts

Ecrire un algorithme qui lit au clavier l'heure, les minutes et les secondes et qui affiche l'heure qu'il sera une seconde plus tard. Par exemple : si l'utilisateur tape 21 puis 32 puis 12 , l'algorithme doit répondre : "Dans une secondes, il sera 21 heure(s) 32 minute(s) 13 seconde(s) ". Utilisez un types structuré.

Exercice 3 : 2.5 pts

Ecrire la procédure *nombre* qui n'a aucun paramètre en entrée. Cette procédure demande à l'utilisateur de saisir un nombre entre 10 et 20, jusqu'à ce que la réponse convienne. En cas de réponse supérieur à 20, on affiche un message : "Plud petit !" et inversement "Plus grand !" si le nombre est inférieur à 10.

Ecole préparatoire en Sciences et Techniques d'Oran
Intitulé : Algorithmique 2 et structures de données avancées — Sigle : Algo2
2^{me} année, Semestre 3, 2011/2012 — Corrigé type du devoir surveillé
Décembre 2011 — Durée : 02h00mn

Partie cours : 10pts

Exercice 1 : (1+1.5+1.5+0.5=4.5pts)

1. la valeur de x=-10
2. La syntaxe générale de l'allocation dynamique en C++ est : *pointeur* = *new type* ;
3. les trois instructions affichent :

   5

   5

   8
4. Les différents cas d'appels corrects de la fonction g sont :
   (a) g(1) ;
   (b) g(1,false) ;
   (c) g(1,false,3.0) ;
5. La compilation séparée en programmation modulaire permet de générer les fichiers objets.

Exercice 2 : 0.5+(0.5+0.5)+ (0.25+0.25*7)+(0.25+0.25*7) = 5.5pts

1. L a fonction h permet de permuter deux valeurs entières si $a < b$. Autrement dit, elle permet de mettre deux valeurs dans l'ordre croissant.
2. On va avoir : i=6 et j=-23. Ceci est du au fait que le passage de paramètres est par valeur, ce qui ne permet pas de modifier les contenus des paramètres d'entrées.
3. Les deux solutions possibles sont :
   – le passage par adresse (solution du C).Ce qui donne :

```
void h(int * a, int * b);
main ()
{
 int i(6),j(-23);
 h(&i,&j);
 cout << "après application de h, i=" << i << "et j=" << j << endl;
}
void h(int * a, int * b)
{
 if (*a>*b)
    {
        *a=*a+*b;
        *b=*a-*b;
        *a=*a-*b;
    }
}
```

   – le passage par référence (solution du C++).

```
void h(int & a, int & b);
main ()
{
 int i(6),j(-23);
 h(i,j);
 cout << "après application de h, i=" << i << "et j=" << j << endl;
}
void h(int & a, int & b)
{
```

```
        if (a>b)
           {
               a=a+b;
               b=a-b;
               a=a-b;
           }
       }
```

Partie TD : 10pts

Exercice 1 : 2pts + 0.5 pts

La fonction *fct* renvoie VRAI si et seulement si la matrice possède deux lignes identique. Il faut justifier en donnant une trace de la fonction (ex : N=5).

Exercice 2 : 5 pts

```
Algorithme Temp2
variable heure : article (0.25 pts)
                      {
                        h:entier naturel; m:entier naturel; s:entier naturel; (0.75pts)
                      };
Début
    écrire("Entrez les heures puis les minutes puis les secondes : "); (0.25pts *5)
    lire(heure.h);
    lire(heure.m);
    lire(heure.s);
    heure.s <-- heure.s+1;
    si heure.s = 60 alors  (0.25pts *3)
                           heure.s<-- 0;
                           heure.m<-- heure.m+1;
    finsi;
    si heure.m = 60 alors (0.25pts *3)
                           heure.m<-- 0;
                           heure.h<-- heure.h+1;
    finsi;
    si heure.h = 24 alors (0.25pts *2)
                           heure.h<-- 0;
    finsi;
    écrire("Dans une seconde il sera ", heure.h, " heure(s) ", heure.m, " minute(s) ",  (0.25pts *3)
            heure.s, " seconde(s) ");
Fin
```

Exercice 3 : 2.5 pts

```
procedure( ) (0.25 pts)
variable N : entier; (0.25 pts)
début
    écrire("Entrez un nombre compris entre 10 et 20 : "); lire(N); (0.25 pts * 2)
    Tant que (N < 10 ou N >20)  (0.5 pts )
    faire
        si (N < 10) alors écrire("Plus grand !") (0.25 pts * 3)
                        sinon écrire("Plus petit !")
        finsi
        lire(N); (0.25 pts)
    fintq
fin
```