

### Question 1 - Tri de formes géométriques (8 points)

Dans cette question, les fonctions `main` et `affiche` n'étaient pas demandées.

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

const double PI = 3.141596;
const int MAX_FIG = 4;

void swap(int forme1[3], int forme2[3])
{
    for(int i=0; i<3;i++)
    {
        int save = forme1[i];
        forme1[i]= forme2[i];
        forme2[i]=save;
    }
}

double aire(int forme[3])
{
    double res = -1; //renvoie -1 si forme[0] différent de 0,1 ou 2
    if(forme[0]==0)
        res = forme[1]*forme[2];
    else if(forme[0]==1)
        res = forme[1]*forme[2]/2.0;
    else if(forme[0]==2)
        res = PI*forme[1]*forme[1];
    return res;
}

void triFormes(int formes[][3])
{
    for(int i = 0; i<MAX_FIG-1; i++)
    {
        int min = i;
        for(int j=i; j<MAX_FIG;j++)
        {
            if(aire(formes[j])< aire(formes[min]))
                min = j;
        }
        swap(formes[i], formes[min]);
    }
}

void affiche(int formes[][3])
{
    for(int i = 0; i<MAX_FIG;i++)
        cout << formes[i][0] << " "
            << formes[i][1] << " "
            << formes[i][2] << endl;
    cout << endl;
}
```

```

int main()
{
    int formes[4][3] = { {2,5,-1},
                        {0,5,10},
                        {1,6,10},
                        {2,3,-1}
                    };

    cout << "Formes avant tri:" << endl;
    affiche(formes);
    triFormes(formes);
    cout << "Formes après tri:" << endl;
    affiche(formes);
    return 0;
}

```

## Question 2 - Equivalence (6 points)

```

const int SIZE = 128;
const char EMPTY_CHAR = ' ';

bool equivalence(char v1[], char v2[])
{
    int i=0,j=0;
    bool ok = true;
    while(i<SIZE && v1[i]==EMPTY_CHAR) i++;
    while(j<SIZE && v2[j]==EMPTY_CHAR) j++;
    while(i<SIZE && j<SIZE && ok)
    {
        if(v1[i]!=v2[j])
            ok = false;
        else
        {
            if(i<SIZE) i++;
            if(j<SIZE) j++;
        }
        while(i<SIZE && v1[i]==EMPTY_CHAR) i++;
        while(j<SIZE && v2[j]==EMPTY_CHAR) j++;
    }
    return (i==SIZE && j==SIZE && ok);
}

```

### Question 3 - Série numérique (6 points)

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

const double EPS = 1.0e-6;
const int N_MAX = 100;

double pow(double base, double exp)
{
    double res = 1;
    for(int i = 0; i < exp; i++)
    {
        res = res * base;
    }
    return res;
}

double abs(double x)
{
    if(x<0)
        x = -x;
    return x;
}

double W(double x)
{
    double factN = 1;
    double xN = x;
    int n = 1;
    double term = x;
    double sum = term;

    while(abs(term) >= EPS && n < N_MAX)
    {
        n++;
        xN = xN * x;
        factN = factN * n;
        term = (pow(-n, n-1)*xN)/factN;
        sum += term;
    }
    return sum;
}

int main()
{
    double x;
    cout << "Entrez une valeur pour x" << endl;
    cin >> x;
    cout << "W(x) = " << W(x) << endl;
    return 0;
}
```