

### Question 1 - Tri Lexicographique (8 points)

```
const int LGMAX = 256;

void copie(char ch1[], const char ch2[])
{
    for(int k=0; k <= ch2[0]; ++k)
        ch1[k] = ch2[k];
}

bool plusPetit(const char ch1[], const char ch2[])
{
    int k=1;
    while(k <= ch1[0] && k <= ch2[0] && ch2[k] == ch1[k])
        ++k;
    return k > ch1[0] || (k <= ch2[0] && ch1[k] < ch2[k]);
}

// Insère val dans tab de [0, nb[
void insere(char val[LGMAX], char tab[][LGMAX], int nb)
{
    int k;
    for(k = nb - 1; k >= 0 && plusPetit(val, tab[k]); --k)
        copie(tab[k + 1], tab[k]);
    copie(tab[k + 1], val);
}

void triTexte(char tab[][LGMAX], int nb)
{
    char tmp[LGMAX];
    for(int i = 1; i < nb; ++i)
    {
        copie(tmp, tab[i]);
        insere(tmp, tab, i);
    }
}
```

### Question 2 - Tirage Aléatoire (6 points)

```
int alea(int deb, int fin)
{
    return rand() % (fin - deb) + deb;
}

char tirerLettre(char sac[], int debut, int& nbLettre)
{
    int pos = alea(debut, debut+nbLettre);
    char lettre = sac[pos];
    sac[pos] = sac[debut + nbLettre - 1];
    --nbLettre;
    return lettre;
}

char tireVoyelle(char sac[], int& nbV, int nbC)
{
    char lettre = tirerLettre(sac,0,nbV);
    sac[nbV] = sac[nbV + nbC];
    return lettre;
}
```

```
char tireConsonne(char sac[], int nbV, int& nbC)
{
    return tirerLettre(sac, nbV, nbC);
}
```

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

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

const double EPS = 1.0e-4;
const int BORNE = 1000;

int euler(int n);
double sech(double x);

double sech(double x)
{
    int signe = 1;
    double terme = 1;
    double somme = 1;
    double x2 = x*x;
    double x2N = 1;
    int fact2N = 1;

    for(int i=1; i<BORNE && terme >= eps; i++)
    {
        signe = - signe;
        x2N = x2N*x2;
        fact2N = fact2N*((2*i)-1)*(2*i);

        terme = ((euler(2*i)/fact2N)*x2N);
        somme += terme*signe;
    }
    return somme;
}

int main()
{
    int x;
    cin >> x;
    cout << sech(x);
    return 0;
}
```