

```

#include <stdio.h>
#include <math.h>

// Fonction a integrer et sa primitive
double f(double x){return sin(x);}
double F(double x){return -cos(x);}

// Declaration des autres fonctions que nous allons utiliser
double rectangles(double a,double b,long int n);
double trapezes (double a,double b,long int n);
double simpson (double a,double b,long int n);

// =====
//                               Programme principal
// =====
main(){

    FILE *unit;
    int n ;                               /* nombre d'iterations */
    long int nmax=100000000; /* Nombre maximal d'iterations */
    double a=0,b=M_PI/2.; /* bornes d'integration */
    double h; /* pas d'integration */
    double I0=F(b)-F(a); /* valeur de l'integrale exacte */
    double I1,I2,I3; /* valeurs des integrales numeriques */

    // On ouvre le fichier
    unit=fopen("integration.dat","w");

    // On calcule les integrales numeriques pour un nombre de points variable
    for(n=2;n<nmax;n*=2){

        fprintf(unit,"%10ld ",n);
        I1=rectangles(a,b,n); fprintf(unit,"%21.13e ",fabs(1.-I1/I0));
        I2=trapezes (a,b,n); fprintf(unit,"%21.13e ",fabs(1.-I2/I0));
        I3=simpson (a,b,n); fprintf(unit,"%21.13e ",fabs(1.-I3/I0));
        fprintf(unit,"\n");

    }

    fclose(unit);
}
// =====

// =====
//                               Calcul de l'integrale par la methode des rectangles
// =====
double rectangles(double a,double b,long int n){

    int k; /* indice de chaque intervalle */
    double h=(b-a)/n; /* pas d'integration */
    double x; /* variable d'integration */
    double I=0; /* valeur de l'integrale */

    //Calcul de l'integral
    x = a ;
    for(k=0;k<n;k++){
        I = I + f(x);
        x = x + h;
    }

    // La multiplication par h est gardee ici pour gagner du temps:
    I = I * h;

    return I;
}
// =====

```

```

// =====
//                               Calcul de l'integrale par la methode des trapezes
// =====
double trapezes(double a,double b,long int n){

    int k; /* indice de chaque intervalle */
    double h=(b-a)/n; /* pas d'integration */
    double x; /* variable d'integration */
    double I=0; /* valeur de l'integrale */

    // Le cas x=a est traite separement
    I=f(a)/2.;

    // Calcul de l'integrale (sans les deux extremes)
    x = a + h ;
    for(k=1;k<n;k++){
        I = I + f(x);
        x = x + h;
    }

    // Le cas x=b est traite separement
    I = I + f(b)/2.;

    // La multiplication par h est gardee ici pour gagner du temps:
    I = I * h;

    return I;
}
// =====

// =====
//                               Calcul de l'integrale par la methode de Simpson
// =====
double simpson(double a,double b,long int n){

    int k; /* indice de chaque intervalle */
    double h=(b-a)/n; /* pas d'integration */
    double x; /* variable d'integration */
    double I=0; /* valeur de l'integrale */

    // Le cas x=a est traite separement
    I=f(a);

    // Calcul de tous les termes impairs de l'integrale
    x=a+h;
    for(k=1;k<n;k=k+2){
        I = I + 4*f(x);
        x = x + 2*h;
    }

    // Calcul de tous les termes pairs de l'integrale
    x=a+2*h;
    for(k=2;k<n;k=k+2){
        I = I + 2*f(x);
        x = x + 2*h;
    }

    /* Le cas x=b est traite separement */
    I = I + f(b);

    // La multiplication par h/3 est gardee ici pour gagner du temps:
    I = I * h/3.;

    return I;
}
// =====

```