```
/* Bisection method */

/*
Algorithm Bisection
1. Read a, b, T
2. Compute f1 = f(a) and f2= f(b)
3. If f1*f2 > 0 then
     i) Read a, b
     ii) Goto step 2
4. Set mid = (a+b)/2 and f3 = f(mid)
5. If (a-b) < T OR f3 = 0 then Print mid and Exit
6. if f1*f3 < 0 then b = mid
     Else f2*f3 < 0 then a = mid
7. Goto step 4
8. End

*/


#include <math.h>
#include <stdio.h>
#include<stdlib.h>
#define MAX_ITER 40

double fof(double x){return(x*x*x - x*x - 1);}

main()
{
   double a,b,mid, f_a,f_b,f_mid;
   double tolerance;

   int iteration;

   do{
   printf("enter the range\n");
   scanf("%lf%lf",&a,&b);

   f_a = fof(a);
   f_b = fof(b);

   }while(f_a*f_b>0.0);

   iteration = 0;
   tolerance = .0001;

   printf("i      a        b          mid        f_of_a      f_of_b      f_mid    \n");

   do{

     mid = 0.5 * (b+a);
     f_mid = fof(mid);

     printf("%d %1.4lf %1.4lf %1.4lf %1.4lf %1.4lf %1.4lf \n",
                          iteration,a,b,mid,f_a,f_b,f_mid);

       if (f_a * f_mid < 0.0) {
             b = mid;
```

```
            f_b = f_mid;
        }
        else {
            a = mid;
            f_a = f_mid;
        }

        f_mid = fof(mid);

        iteration++;

    }while(fabs(b-a)>tolerance && iteration < MAX_ITER);

    if(iteration < MAX_ITER)
    printf("\nSolution is x = %lf function = %lf iterations = %d    \n\n\n", mid,f_mid,iteration);
    else printf("\nApproximations does not converge");
}
```

```
/*
Algorithm Eular
1. Read x, y, h, t
2. Repeat while x <= t
        a. Set k=h*fun(x,y);
        b. Set y=y+k;
        c. Set x=x+h;
        [End of loop]
3. Print y
4. End

*/

#include<stdio.h>
float fun(float x,float y){return x+y;}

main()
{
        float a,b,x,y,h,t,k;
        printf("\nEnter x0,y0,h,xn: ");
        scanf("%f%f%f%f",&x,&y,&h,&t);

        printf("\n    x\t    y\n");
        while(x<=t)
        {
                k=h*fun(x,y);
                y=y+k;
                x=x+h;
                printf("%0.3f\t%0.3f\n",x,y);
        }
}
```

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

#define MAX 50
#define T 0.000001

double fof(double x){return(x*x*x - x - 1.0);}
double dof(double x){return(3*x*x - 1.0);}

main()
{
   double fx, df, dx, rtn;
   int j;

  printf("Enter your guess");
  scanf("%lf",&rtn);

  printf("Iteration              root\n");

for (j=1;j<=MAX; j++)
{
   fx = fof(rtn);
   df = dof(rtn);
   dx = -fx/df;
   rtn += dx;
   printf("%d                     %f\n", j,rtn);

   if (fabs(dx) < T){
       printf("found root after %d attempts, at %lf\n", j, rtn);
       exit(0);
       }
}
  printf("error - exceeded max tries no root");

}
```

```
/*
Algorithm Simpson_one_third
1. Read a, b, n
2. Set h = (b-a)/n
3. Repeat For I=1 to n-1 do
     a. Set x=a+i*h
     b. If I mod 2 = 0 then
          Set sum=sum+2*f(x)
          Else
          sum=sum+4*f(x);
     [End of loop]
4. Set sum =(h/3)*(f(a)+f(b)+sum)
5. Print sum
6. End
*/

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

double f(double x){ return 1/(1+x*x);}

main()
{
   int c, i, n;
   float a, b, h, sum;

   printf("\nEnter lower limit, upper limit and number of subinterval\n");
   scanf("%f%f%d", &a,&b,&n);

   h=(b-a)/n;                          /* Step size h=(b-a)/N */
   c=2;                                  /* gives the 4,2,4,2,... in the algorithm       */
   for(i=1;i<n;i++)
   {
      c=6-c;
      sum=sum+c*f(a+i*h);
   }
   sum=(h/3)*(f(a)+f(b)+sum);
     /*Print the answer */
   printf("\nThe integral is: %lf\n",sum);
}
```