**Q1**

1. **Taylor Polynomial in C:**

#include<stdio.h>

#include<math.h>

int main()

{

printf("\n\n\t\tStudytonight - Best place to learn\n\n\n");

int x,i;

int fact = 1,n;

float sum=0;

printf("\n\nEnter the value of x in the series : ");

scanf("%d",&x);

printf("\nEnter the number of terms in the series : ");

scanf("%d",&n);

for(i=1;i<n;i++)

{

fact = fact\*i;

sum = sum + (pow(x,i)/fact) ;

}

sum= sum +1; //Since series starts with 1

printf("\n\nThe sum of the taylor series is : %.2f\n\n",sum);

printf("\n\n\n\n\t\t\tCoding is Fun !\n\n\n");

return 0;

}

1. **Relative Error in C:**

#include<stdio.h>  
main()  
{

float av,n,tv,t1,t2,e,ea,er,ep,temp;  
printf(“Enter true value”);  
scanf(“%f”,&tv);  
n=tv;  
t1=(n\*10000);  
t2=(int)t1%10;  
if(t2>5)  
{  
n=n+0.001;

}

if(t2==5)  
{  
t1=(n\*1000);  
t2=(int)t1%10;  
if((int)t2%2!=0)  
n=n+0.001;

}  
printf(“%f”,n);  
t1=n\*1000;  
t1=t1-(int)t1;  
t1=t1\*.001;  
n=n-t1;  
e=tv-n;  
ea=fabs(e);  
er=ea/tv;  
ep=er\*100;  
printf(“\nabsolute value…%f”,n);  
printf(“\nerror…%f”,e);  
printf(“\nabsolute error…%f”,ea);  
printf(“\nrelative error…%f”,er);  
printf(“\npercentage error….%f”,ep);

}

**Q2**

**Bisection Method in C:**

#include<stdio.h>

//function used is x^3-2x^2+3

double func(double x)

{

    return x\*x\*x - 2\*x\*x + 3;

}

double e=0.01;

double c;

void bisection(double a,double b)

{

    if(func(a) \* func(b) >= 0)

    {

        printf("Incorrect a and b");

        return;

    }

    c = a;

    while ((b-a) >= e)

    {

        c = (a+b)/2;

        if (func(c) == 0.0){

            printf("Root = %lf\n",c);

            break;

        }

        else if (func(c)\*func(a) < 0){

                printf("Root = %lf\n",c);

                b = c;

        }

        else{

                printf("Root = %lf\n",c);

                a = c;

        }

    }

}

int main()

{

    double a,b;

    a=-10;

    b=20;

    printf("The function used is x^3-2x^2+3\n");

    printf("a = %lf\n",a);

    printf("b = %lf\n",b);

    bisection(a,b);

    printf("\n");

    printf("Accurate Root calculated is = %lf\n",c);

    return 0;

}

**Fixed Point in C:**

#include<stdio.h>

#include<conio.h>

#include<math.h>

/\* Define function f(x) which

is to be solved \*/

#define f(x) cos(x)-3\*x+1

/\* Write f(x) as x = g(x) and

define g(x) here \*/

#define g(x) (1+cos(x))/3

int main()

{

int step=1, N;

float x0, x1, e;

clrscr();

/\* Inputs \*/

printf("Enter initial guess: ");

scanf("%f", &x0);

printf("Enter tolerable error: ");

scanf("%f", &e);

printf("Enter maximum iteration: ");

scanf("%d", &N);

/\* Implementing Fixed Point Iteration \*/

printf("\nStep\tx0\t\tf(x0)\t\tx1\t\tf(x1)\n");

do

{

x1 = g(x0);

printf("%d\t%f\t%f\t%f\t%f\n",step, x0, f(x0), x1, f(x1));

step = step + 1;

if(step>N)

{

printf("Not Convergent.");

exit(0);

}

x0 = x1;

}while( fabs(f(x1)) > e);

printf("\nRoot is %f", x1);

getch();

return(0);

}

**Newtons Method in C:**

#include<stdio.h>

#include<conio.h>

#include<math.h>

#include<stdlib.h>

/\* Defining equation to be solved.

Change this equation to solve another problem. \*/

#define f(x) 3\*x - cos(x) -1

/\* Defining derivative of g(x).

As you change f(x), change this function also. \*/

#define g(x) 3 + sin(x)

void main()

{

float x0, x1, f0, f1, g0, e;

int step = 1, N;

clrscr();

/\* Inputs \*/

printf("\nEnter initial guess:\n");

scanf("%f", &x0);

printf("Enter tolerable error:\n");

scanf("%f", &e);

printf("Enter maximum iteration:\n");

scanf("%d", &N);

/\* Implementing Newton Raphson Method \*/

printf("\nStep\t\tx0\t\tf(x0)\t\tx1\t\tf(x1)\n");

do

{

g0 = g(x0);

f0 = f(x0);

if(g0 == 0.0)

{

printf("Mathematical Error.");

exit(0);

}

x1 = x0 - f0/g0;

printf("%d\t\t%f\t%f\t%f\t%f\n",step,x0,f0,x1,f1);

x0 = x1;

step = step+1;

if(step > N)

{

printf("Not Convergent.");

exit(0);

}

f1 = f(x1);

}while(fabs(f1)>e);

printf("\nRoot is: %f", x1);

getch();

}

**Secant Method in C:**

/\* Program: Finding real roots of nonlinear

equation using Secant Method

Author: CodeSansar

Date: November 18, 2018 \*/

#include<stdio.h>

#include<conio.h>

#include<math.h>

#include<stdlib.h>

/\* Defining equation to be solved.

Change this equation to solve another problem. \*/

#define f(x) x\*x\*x - 2\*x - 5

void main()

{

float x0, x1, x2, f0, f1, f2, e;

int step = 1, N;

clrscr();

/\* Inputs \*/

printf("\nEnter initial guesses:\n");

scanf("%f%f", &x0, &x1);

printf("Enter tolerable error:\n");

scanf("%f", &e);

printf("Enter maximum iteration:\n");

scanf("%d", &N);

/\* Implementing Secant Method \*/

printf("\nStep\t\tx0\t\tx1\t\tx2\t\tf(x2)\n");

do

{

f0 = f(x0);

f1 = f(x1);

if(f0 == f1)

{

printf("Mathematical Error.");

exit(0);

}

x2 = x1 - (x1 - x0) \* f1/(f1-f0);

f2 = f(x2);

printf("%d\t\t%f\t%f\t%f\t%f\n",step,x0,x1,x2, f2);

x0 = x1;

f0 = f1;

x1 = x2;

f1 = f2;

step = step + 1;

if(step > N)

{

printf("Not Convergent.");

exit(0);

}

}while(fabs(f2)>e);

printf("\nRoot is: %f", x2);

getch();

}

**False Posistion Method in C:**

#include<stdio.h>

#include<conio.h>

#include<math.h>

/\* Defining equation to be solved.

Change this equation to solve another problem. \*/

#define f(x) x\*log10(x) - 1.2

int main()

{

float x0, x1, x2, f0, f1, f2, e;

int step = 1;

clrscr();

/\* Inputs \*/

up:

printf("\nEnter two initial guesses:\n");

scanf("%f%f", &x0, &x1);

printf("Enter tolerable error:\n");

scanf("%f", &e);

/\* Calculating Functional Values \*/

f0 = f(x0);

f1 = f(x1);

/\* Checking whether given guesses brackets the root or not. \*/

if( f0\*f1 > 0.0)

{

printf("Incorrect Initial Guesses.\n");

goto up;

}

/\* Implementing Regula Falsi or False Position Method \*/

printf("\nStep\t\tx0\t\tx1\t\tx2\t\tf(x2)\n");

do

{

x2 = x0 - (x0-x1) \* f0/(f0-f1);

f2 = f(x2);

printf("%d\t\t%f\t%f\t%f\t%f\n",step, x0, x1, x2, f2);

if(f0\*f2 < 0)

{

x1 = x2;

f1 = f2;

}

else

{

x0 = x2;

f0 = f2;

}

step = step + 1;

}while(fabs(f2)>e);

printf("\nRoot is: %f", x2);

getch();

return 0;

}

**Q3**

**Lagrange Interpolation in C:**

#include<stdio.h>

main()

{

float x[100],y[100],a,s=1,t=1,k=0;

int n,i,j,d=1;

printf("\n\n Enter the number of the terms of the table: ");

scanf("%d",&n);

printf("\n\n Enter the respective values of the variables x and y: \n");

for(i=0; i<n; i++)

{

scanf ("%f",&x[i]);

scanf("%f",&y[i]);

}

printf("\n\n The table you entered is as follows :\n\n");

for(i=0; i<n; i++)

{

printf("%0.3f\t%0.3f",x[i],y[i]);

printf("\n");

}

while(d==1)

{

printf(" \n\n\n Enter the value of the x to find the respective value of y\n\n\n");

scanf("%f",&a);

for(i=0; i<n; i++)

{

s=1;

t=1;

for(j=0; j<n; j++)

{

if(j!=i)

{

s=s\*(a-x[j]);

t=t\*(x[i]-x[j]);

}

}

k=k+((s/t)\*y[i]);

}

printf("\n\n The respective value of the variable y is: %f",k);

printf("\n\n Do you want to continue?\n\n Press 1 to continue and any other key to exit");

scanf("%d",&d);

}

}