**CODING IN ‘C’**

1. **Calculate the sum 1/1 + 1/2 + 1/3 + 1/4 + …. + 1/ N.**

***/\****

***\* C Program to find the Sum of Series 1 + 1/2 + 1/3 + 1/4 + ... + 1/N***

***\*/***

**#include <stdio.h>**

**void main()**

**{**

**double number, sum = 0, i;**

**printf("\n enter the number ");**

**scanf("%lf", &number);**

**for (i = 1; i <= number; i++)**

**{**

**sum = sum + (1 / i);**

**if (i == 1)**

**printf("\n 1 +");**

**else if (i == number)**

**printf(" (1 / %lf)", i);**

**else**

**printf(" (1 / %lf) + ", i);**

**}**

**printf("\n The sum of the given series is %.2lf", sum);**

**}**

1. **Enter 100 integers into an array and sort them in an ascending order.**

***\* C program to accept N numbers and arrange them in an ascending order***

***\*/***

**#include <stdio.h>**

**void main()**

**{**

**int i, j, a, n, number[30];**

**printf("Enter the value of N \n");**

**scanf("%d", &n);**

**printf("Enter the numbers \n");**

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

**scanf("%d", &number[i]);**

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

**{**

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

**{**

**if (number[i] > number[j])**

**{**

**a = number[i];**

**number[i] = number[j];**

**number[j] = a;**

**}**

**}**

**}**

**printf("The numbers arranged in ascending order are given below \n");**

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

**printf("%d\n", number[i]);**

**}**

1. **a)Program for Bisection Method in C**

## ****#include<stdio.h>**** ****#include<conio.h>**** ****#include<math.h>**** ****#include<stdlib.h>**** ****float f(float x)**** ****{**** ****float sum;**** ****sum=pow(x,3)+x+3;**** ****return sum;**** ****}**** ****main()**** ****{**** ****float a,b,c,y0,y1,m;**** ****step:**** ****printf(“enter the value of a,b and c “);**** ****scanf(“%f%f%f”,&a,&b,&c);**** ****y1=f(a);**** ****y0=f(b);**** ****if(y0\*y1>0)**** ****{**** ****goto step;**** ****}**** ****step1:**** ****m=(a+b)/2;**** ****if(f(m)==0)**** ****{**** ****printf(“the root is : %f”,m);**** ****exit(0);**** ****}**** ****else**** ****{**** ****if (y1\*f(m)<0)**** ****b=m;**** ****else**** ****a=m;**** ****}**** ****if(fabs(a-b)>c)**** ****{**** ****goto step1;**** ****}**** ****printf(“the root is :%f”,m);**** ****getch();**** ****}****

**3d) Write a C program and run to find the smallest positive root of the equation *tanx +a sin2x=1, a=.3* By Newton Raphson Method correct upto 5 significant figure.**

PROGRAMME

/\* Solution by Newton Raphson Method \*/

#include <stdio.h>

#include <conio.h>

#include <math.h>

#define err 0.00001

#define ITNO 20

#define F(x) tan(x)+a\* pow(sin(x),2)-1

#define FD(x) (1/pow(cos(x),2))+a\*sin(2\*x)

void main( )

{

int count;

float x0,xn,fx,fdx,a=.3;

printf(" give the initial approximation \n");

scanf("%f",&x0);

count=1;

begin:

fx=F(x0);

fdx=FD(x0);

xn=x0-(fx/fdx);

printf( "\n n=%d xn=%f",count,xn);

if(fabs(xn-x0)<err)

printf(" \nthe root is %f",xn);

else

{

x0=xn;

count=count+1;

if(count<=ITNO)

{

goto begin;

}

else

{

printf( "the solution does not converge");

}

}

getch();

}

**4c) Code for JACOBIAN METHOD in C Programming**

**#include<stdio.h>**

**#include<conio.h>**

**#include<math.h>**

**#define ESP 0.0001**

**#define X1(x2,x3) ((17 - 20\*(x2) + 2\*(x3))/20)**

**#define X2(x1,x3) ((-18 - 3\*(x1) + (x3))/20)**

**#define X3(x1,x2) ((25 - 2\*(x1) + 3\*(x2))/20)**

**void main()**

**{**

**double x1=0,x2=0,x3=0,y1,y2,y3;**

**int i=0;**

**clrscr();**

**printf("\n\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n");**

**printf("\n x1\t\t x2\t\t x3\n");**

**printf("\n\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n");**

**printf("\n%f\t%f\t%f",x1,x2,x3);**

**do**

**{**

**y1=X1(x2,x3);**

**y2=X2(x1,x3);**

**y3=X3(x1,x2);**

**if(fabs(y1-x1)<ESP && fabs(y2-x2)<ESP && fabs(y3-x3)<ESP )**

**{**

**printf("\n\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n");**

**printf("\n\nx1 = %.3lf",y1);**

**printf("\n\nx2 = %.3lf",y2);**

**printf("\n\nx3 = %.3lf",y3);**

**i = 1;**

**}**

**else**

**{**

**x1 = y1;**

**x2 = y2;**

**x3 = y3;**

**printf("\n%f\t%f\t%f",x1,x2,x3);**

**}**

**}while(i != 1);**

**getch();**

**}**

/\*

## 5a) Source Code for 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)**

**{**

**k=0;**

**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);**

**}**

**}**

**6b) Write a C program and run to Evaluate , a=.8 by Simson’s Rule correct upto 5 significant figures.**

PROGRAMME

/\*Simson\*/

#include <stdio.h>

#include <conio.h>

#include <math.h>

void main( )

{

int n,i;

float a,b,h,sum1,sum2,sum3,intvle;

float F(float x);

printf(" give the initial value of limit a\n");

scanf("%f",&a);

printf( "give the value of final limit b");

scanf("%f",&b);

printf( "give the value of interval 'n'");

scanf("%d",&n);

h=(b-a)/n;

sum1=(F(a)+F(b));

sum2=0;

for (i=1;i<n;i=i+2)

{

sum2=sum2+F(a+i\*h);

}

sum3=0;

for (i=2;i<n-1;i=i+2)

{

sum3=sum3+F(a+i\*h);

}

intvle=h\*(sum1+4\*sum2+2\*sum3)/3;

printf("\n");

printf("a..........b............n............Value of Integration");

printf( "\n%f %f %d %5f",a,b,n,intvle);

getch();

}

float F(float x)

{

float f;

f=(log(1+.8\*x+.64))/(.8+x);

return (f);

}

**7) Power Method for Dominant Eigen Values**

**#include<stdio.h>**

**#include<conio.h>**

**#include<math.h>**

**void main()**

**{**

**int i,j,n;**

**float A[40][40],x[40],z[40],e[40],zmax,emax;**

**printf("\nEnter the order of matrix:");**

**scanf("%d",&n);**

**printf("\nEnter matrix elements row-wise\n");**

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

**{**

**for(j=1; j<=n; j++)**

**{**

**printf("A[%d][%d]=", i,j);**

**scanf("%f",&A[i][j]);**

**}**

**}**

**printf("\nEnter the column vector\n");**

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

**{**

**printf("X[%d]=",i);**

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

**}**

**do**

**{**

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

**{**

**z[i]=0;**

**for(j=1; j<=n; j++)**

**{**

**z[i]=z[i]+A[i][j]\*x[j];**

**}**

**}**

**zmax=fabs(z[1]);**

**for(i=2; i<=n; i++)**

**{**

**if((fabs(z[i]))>zmax)**

**zmax=fabs(z[i]);**

**}**

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

**{**

**z[i]=z[i]/zmax;**

**}**

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

**{**

**e[i]=0;**

**e[i]=fabs((fabs(z[i]))-(fabs(x[i])));**

**}**

**emax=e[1];**

**for(i=2; i<=n; i++)**

**{**

**if(e[i]>emax)**

**emax=e[i];**

**}**

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

**{**

**x[i]=z[i];**

**}**

**}**

**while(emax>0.001);**

**printf("\n The required eigen value is %f",zmax);**

**printf("\n\nThe required eigen vector is :\n");**

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

**{**

**printf("%f\t",z[i]);**

**}**

**getch();**

**}**

**9a) Euler’s method in C to solve the ordinary differential equation dy/dx = x+y**.

#include<stdio.h>

float fun(float x,float y)

{

    float f;

    f=x+y;

    return f;

}

main()

{

    float a,b,x,y,h,t,k;

    printf("\nEnter x0,y0,h,xn: ");

    scanf("%f%f%f%f",&a,&b,&h,&t);

    x=a;

    y=b;

    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);

    }

}

**9c) Write a C program and run to find the value of y at x=.4 from the ODE  given that y(0)=1 by Runge Kutta Method correct uoto 3 significant figures.**

PROGRAMME

/\*Runge kutta\*/

#include <stdio.h>

#include <conio.h>

#include <math.h>

void main( )

{

int n,i;

float k1,k2,k3,k4,k,xp,h,x,y;

float g(float,float);

printf("give the value of x0");

scanf("%f",&x);

printf("give the value of xp");

scanf("%f",&xp);

printf("give the value of y0");

scanf("%f",&y);

printf("give the value of h");

scanf("%f",&h);

/\* step length \*/

n=(int)((xp-x)/h);

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

{

k1=h\*g(x,y);

k2=h\*g(x+0.5\*h,y+0.5\*k1);

k3=h\*g(x+0.5\*h,y+0.5\*k2);

k4=h\*g(x+h,y+k3);

k=(k1+2\*k2+2\*k3+k4)/6.0;

x=x+h;y=y+k;

printf( " \n %d %f %f %3f", i,x,y);

}

getch();

}

float g(float x,float y)

{

float f;

f=(x-y)/(x+y);

return(f);

}