**RUNGE KUTTA 4TH ORDER METHOD**

**AND**

**RUNGE KUTTA 2ND ORDER METHOD**

**EXAMPLE 1 : FIND THE VALUE OF Y WHEN X=1. GIVEN THAT Y=1 WHEN**

**X=0. USE THE 4TH ORDER RUNGE KUTTA METHOD.**

**PROGRAM:**

**#include<stdio.h>**

**#include<conio.h>**

**#include<math.h>**

**// function calling**

**float fun(float,float);**

**void main()**

**{**

**FILE \*fin,\*fout;**

**float h,x,y,x0,y0,k1,k2,k3,k4;**

**clrscr();**

**fin=fopen("r1.in","r");**

**fout=fopen("r1.out","w");**

**fscanf(fin,"%f %f %f %f",&x0,&y0,&h,&x);**

**// formulation of rk method**

**S:**

**if(x0==x)**

**goto K;**

**{**

**k1=h\*fun(x0,y0);**

**k2=h\*fun(x0+(h/2),y0+(k1/2));**

**k3=h\*fun(x0+(h/2),y0+(k2/2));**

**k4=h\*fun(x0+h,y0+k3);**

**y=y0+((k1+(2.0\*k2)+(2.0\*k3)+k4)/6.0);**

**x0=x0+h;**

**y0=y;**

**fprintf(fout,"\n%f \t %f\n",x0,y0);**

**goto S;**

**}**

**K:**

**getch();**

**}**

**// defining function**

**float fun(float x,float y)**

**{**

**float f;**

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

**return(f);**

**}**

**Input:**

**0.0**

**1.0**

**0.2**

**1.0**

**Output:**

**0.200000 1.167862**

**0.400000 1.290169**

**0.600000 1.381713**

**0.800000 1.449703**

**1.000000 1.498311**

**EXAMPLE 2: FIND THE VALUE OF Y FOR 1<=t<=6. GIVEN THAT Y(1)=2.**

**USE THE 2ND AND 4TH ORDER RUNGE KUTTA METHOD.**

**=7t2-)**

**PROGRAM : (RUNGE KUTTA 4TH ORDER)**

**#include<stdio.h>**

**#include<conio.h>**

**#include<math.h>**

**// function calling**

**float fun(float,float);**

**void main()**

**{**

**FILE \*fin,\*fout;**

**float h,t,y,t0,y0,k1,k2,k3,k4;**

**clrscr();**

**fin=fopen("r1.in","r");**

**fout=fopen("r1.out","w");**

**fscanf(fin,"%f %f %f %f",&t0,&y0,&h,&t);**

**// formulation of rk method**

**S:**

**if(t0==t)**

**goto K;**

**{**

**k1=h\*fun(t0,y0);**

**k2=h\*fun(t0+(h/2),y0+(k1/2));**

**k3=h\*fun(t0+(h/2),y0+(k2/2));**

**k4=h\*fun(t0+h,y0+k3);**

**y=y0+((k1+(2.0\*k2)+(2.0\*k3)+k4)/6.0);**

**t0=t0+h;**

**y0=y;**

**fprintf(fout,"\n%f \t %f\n",t0,y0);**

**goto S;**

**}**

**K:**

**getch();**

**}**

**// defining function**

**float fun(float t,float y)**

**{**

**float f;**

**f=(7.0\*t\*t)-(4.0\*y/t);**

**return(f);**

**}**

**Input :**

**1.0**

**2.0**

**1.0**

**6.0**

**Output :**

**2.000000 9.750000**

**3.000000 27.846666**

**4.000000 64.491074**

**5.000000 125.337669**

**6.000000 216.256317**

**PROGRAM : (RUNGE KUTTA 2ND ORDER)**

**#include<stdio.h>**

**#include<conio.h>**

**#include<math.h>**

**float fun(float,float);**

**void main()**

**{**

**FILE \*fin,\*fout;**

**float x0,y0,h,k1,k2,k3,k4,x,y;**

**clrscr();**

**fin=fopen("rk2.in","r");**

**fout=fopen("rk2.out","w");**

**fscanf(fin,"%f%f%f%f",&x0,&y0,&h,&x);**

**S:**

**if (x0==x)**

**goto K;**

**{**

**k1=h\*fun(x0,y0);**

**k2=h\*fun(x0+h,y0+k1);**

**y=y0+(k1+k2)/2.0;**

**x0=x0+h;**

**y0=y;**

**fprintf(fout,"\n%f\t%f\n",x0,y0);**

**goto S;**

**}**

**K:**

**getch();**

**}**

**float fun(float x,float y)**

**{**

**float f;**

**f=((7\*x\*x)-(4\*y/x));**

**return(f);**

**}**

**INPUT**

**1.0**

**2.0**

**1.0**

**6.0**

**OUTPUT**

**2.0000 14.50000**

**3.0000 36.50000**

**4.0000 74.25000**

**5.0000 135.8249**

**6.0000 227.6066**

**EXAMPLE 3: USE THE 4TH ORDER RUNGE KUTTA METHOD TO SOLVE**

**10=x2+y2**

**WHERE Y(0)=1 FOR THE RANGE OF 0.00<=X<=0.4,h=0.1**

**PROGRAM:**

**#include<stdio.h>**

**#include<conio.h>**

**#include<math.h>**

**// function calling**

**float fun(float,float);**

**void main()**

**{**

**FILE \*fin,\*fout;**

**float h,x,y,x0,y0,k1,k2,k3,k4;**

**clrscr();**

**fin=fopen("r1.in","r");**

**fout=fopen("r1.out","w");**

**fscanf(fin,"%f %f %f %f",&x0,&y0,&h,&x);**

**// formulation of rk method**

**S:**

**if(x0==x)**

**goto K;**

**{**

**k1=h\*fun(x0,y0);**

**k2=h\*fun(x0+(h/2),y0+(k1/2));**

**k3=h\*fun(x0+(h/2),y0+(k2/2));**

**k4=h\*fun(x0+h,y0+k3);**

**y=y0+((k1+(2.0\*k2)+(2.0\*k3)+k4)/6.0);**

**x0=x0+h;**

**y0=y;**

**fprintf(fout,"\n%f \t %f\n",x0,y0);**

**goto S;**

**}**

**K:**

**getch();**

**}**

**// defining function**

**float fun(float x,float y)**

**{**

**float f;**

**f=(x\*x+y\*y)/10.0;**

**return(f);**

**}**

**Input:**

**0.0**

**1.0**

**0.1**

**0.4**

**Output:**

**0.100000 1.010134**

**0.200000 1.020678**

**0.300000 1.031842**

**0.400000 1.043845**