```
/**********************************************************************************
                                   Euler's Method to solve first order ODE
**********************************************************************************/


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

float f(float x, float y){
return x*x+y*y;
}

int main(){
    int i, n;
    float x, y, xp, h, dy;
    printf("\nInput initial values of x and y: ");
    scanf("%f%f", &x, &y);
    printf("\nInput x at which y is required: ");
    scanf("%f", &xp);
    printf("\nInput step-size h: ");
    scanf("%f", &h);
    n=(int)((xp-x)/h+0.5);
    for(i=1;i<=n;i++){
        dy=h*f(x, y);
        x=x+h;
        y=y+dy;
        printf("\n%d\t  %f\t  %f\n", i, x, y);
    }
    printf("\nValue of y at x = %f is %f\n", x, y);
    getch();
    return 0;
}
```

```
/*******************************************************************************
                          Heun's Method to solve first order ODE
*******************************************************************************/


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

float f(float x, float y){
return y-x*x+1;
}

int main(){
   int i, n;
   float x, y, xp, h, m1, m2;
   printf("\nInput initial values of x and y: ");
   scanf("%f%f", &x, &y);
   printf("\nInput x at which y is required: ");
   scanf("%f", &xp);
   printf("\nInput step-size h: ");
   scanf("%f", &h);
   n=(int)((xp-x)/h+0.5);
   for(i=1;i<=n;i++){
      m1=f(x, y);
      m2=f(x+h, y+m1*h);
      x=x+h;
      y=y+0.5*h*(m1+m2);
      printf("\n%d \t %f \t %f\n", i, x, y);
   }
   printf("\nValue of y at x = %f is %f\n", x, y);
   getch();
   return 0;
}
```

```
/********************************************************************************
                       4th order Runge-Kutta Method to solve first order ODE
********************************************************************************/


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

float f(float x, float y){
return y-x*x+1;
}

int main(){
   int i, n;
   float x, y, xp, h, m1, m2, m3, m4;
   printf("\nInput initial values of x and y: ");
   scanf("%f%f", &x, &y);
   printf("\nInput x at which y is required: ");
   scanf("%f", &xp);
   printf("\nInput step-size h: ");
   scanf("%f", &h);
   n=(int)((xp-x)/h+0.5);
   for(i=1;i<=n;i++){
      m1=f(x, y);
      m2=f(x+0.5*h, y+0.5*m1*h);
      m3=f(x+0.5*h, y+0.5*m2*h);
      m4=f(x+h, y+m3*h);
      x=x+h;
      y=y+(m1+2.0*m2+2.0*m3+m4)*h/6.0;
      printf("\n%d\t %f\t %f", i, x, y);
   }
   printf("\nValue of y at x = %f is %f\n", x, y);
   getch();
   return 0;
}
```

```
/*******************************************************************************************
                         Solving System of Differential Equations
*******************************************************************************************/

#include <stdio.h>
#include <math.h>
#include <conio.h>
#define EPS 0.00001

float f1(float x, float y1, float y2){
        return -4*y1-2*y2+cos(x)+4*sin(x);
}
float f2(float x, float y1, float y2){
        return 3*y1+y2-3*sin(x);
}

//Routine for Euler's method
void euler(float x0, float x1, float y0, float y1, float h){
        float m1, m2;
        printf("\n\nCalculation of y_1(%f) and y_2(%f):", x1, x1);
        printf("\n\n x \t\t y_1(x) \t\t y_2(x)");
        while (fabs(x0-x1)>0.0001){
                printf("\n %f\t%f\t%f", x0, y0, y1);
                m1=f1(x0, y0, y1);
                m2=f2(x0, y0, y1);
                y0=y0+h*m1;
                y1=y1+h*m2;
                x0=x0+h;
        }
        printf("\n %f\t%f\t%f", x0, y0, y1);
        return;
}

//Routine for Heun's method
void heun(float x0, float x1, float y0, float y1, float h){
        float m11, m12, m21, m22;
        printf("\n\nCalculation of y_1(%f) and y_2(%f):", x1, x1);
        printf("\n\n x \t\t y_1(x) \t\t y_2(x)");
        while (fabs(x0-x1)>0.0001){
                printf("\n %f\t%f\t%f", x0, y0, y1);
                m11=f1(x0, y0, y1);
                m21=f2(x0, y0, y1);
                m12=f1(x0+h, y0+h*m11, y1+h*m21);
                m22=f2(x0+h, y0+h*m11, y1+h*m21);
                y0=y0+0.5*h*(m11+m12);
                y1=y1+0.5*h*(m21+m22);
                x0=x0+h;
        }
        printf("\n %f\t%f\t%f", x0, y0, y1);
        return;
}
```

```
//Routine for Runge-Kutta method
void rk(float x0, float x1, float y0, float y1, float h){
        float k1, k2, k3, k4, l1, l2, l3, l4;
        printf("\n\nCalculation of y_1(%f) and y_2(%f):", x1, x1);
        printf("\n\n x \t\t y_1(x) \t\t y_2(x)");
        while (fabs(x0-x1)>0.0001){
                printf("\n %f\t%f\t%f", x0, y0, y1);
                k1=f1(x0, y0, y1);
                l1=f2(x0, y0, y1);
                k2=f1(x0+0.5*h, y0+0.5*h*k1, y1+0.5*h*l1);
                l2=f2(x0+0.5*h, y0+0.5*h*k1, y1+0.5*h*l1);
                k3=f1(x0+0.5*h, y0+0.5*h*k2, y1+0.5*h*l2);
                l3=f2(x0+0.5*h, y0+0.5*h*k2, y1+0.5*h*l2);
                k4=f1(x0+h, y0+h*k3, y1+h*l3);
                l4=f2(x0+h, y0+h*k3, y1+h*l3);
                y0=y0+(h/6)*(k1+2*(k2+k3)+k4);
                y1=y1+(h/6)*(l1+2*(l2+l3)+l4);
                x0=x0+h;
        }
        printf("\n %f\t%f\t%f", x0, y0, y1);
        return;
}

int main()
{
        float x0, yx0, yx1, xp, h;
        char q;
        printf("\n Enter the initial point x: ");
        scanf("%f", &x0);
        printf("\n Enter the value of y_1(x): ");
        scanf("%f", &yx0);
        printf("\n Enter the value of y_2(x): ");
        scanf("%f", &yx1);
        printf("\n Enter the step length: ");
        scanf("%f", &h);
        do{
                printf("\n\n Enter the point x at which y(x) is required: ");
                scanf("%f", &xp);
                heun(x0, xp, yx0, yx1, h);
                ~~//printf("\n The approximate value of y_1(%f) is %f", xp, yxp);~~
                printf("\n\n Do you want to approximate at another point?(y/n): ");
                scanf(" %c", &q);
        } while (q=='y');
        getch();
        return 0;
}
```

```
/**********************************************************************************
                              Solving 2nd Order IVP
**********************************************************************************/

#include <stdio.h>
#include <math.h>
#include <conio.h>
#define EPS 0.00001

float f1(float x, float y1, float y2){
        return y2;
}
float f2(float x, float y1, float y2){
        return 2*y2-y1+x*(exp(x)-1);
}

//Routine for Euler's method
float euler(float x0, float x1, float y0, float y1, float h){
        float m1, m2;
        printf("\n\nCalculation of y(%f):", x1);
        printf("\n x \t\t y(x)");
        while (fabs(x0-x1)>0.0001){
                printf("\n %f\t%f", x0, y0);
                m1=f1(x0, y0, y1);
                m2=f2(x0, y0, y1);
                y0=y0+h*m1;
                y1=y1+h*m2;
                x0=x0+h;
        }
        printf("\n %f\t%f", x0, y0);
        return y0;
}

//Routine for Heun's method
float heun(float x0, float x1, float y0, float y1, float h){
        float m11, m12, m21, m22;
        printf("\n\nCalculation of y(%f):", x1);
        printf("\n x \t\t y(x)");
        while (fabs(x0-x1)>0.0001){
                printf("\n %f\t%f", x0, y0);
                m11=f1(x0, y0, y1);
                m21=f2(x0, y0, y1);
                m12=f1(x0+h, y0+h*m11, y1+h*m21);
                m22=f2(x0+h, y0+h*m11, y1+h*m21);
                y0=y0+0.5*h*(m11+m12);
                y1=y1+0.5*h*(m21+m22);
                x0=x0+h;
        }
        printf("\n %f\t%f", x0, y0);
        return y0;
}
```

```
//Routine for Runge-Kutta method
float rk(float x0, float x1, float y0, float y1, float h){
        float k1, k2, k3, k4, l1, l2, l3, l4;
        printf("\n\nCalculation of y(%f):", x1);
        printf("\n x \t\t y(x)");
        while (fabs(x0-x1)>0.0001){
                printf("\n %f\t%f", x0, y0);
                k1=f1(x0, y0, y1);
                l1=f2(x0, y0, y1);
                k2=f1(x0+0.5*h, y0+0.5*h*k1, y1+0.5*h*l1);
                l2=f2(x0+0.5*h, y0+0.5*h*k1, y1+0.5*h*l1);
                k3=f1(x0+0.5*h, y0+0.5*h*k2, y1+0.5*h*l2);
                l3=f2(x0+0.5*h, y0+0.5*h*k2, y1+0.5*h*l2);
                k4=f1(x0+h, y0+h*k3, y1+h*l3);
                l4=f2(x0+h, y0+h*k3, y1+h*l3);
                y0=y0+(h/6)*(k1+2*(k2+k3)+k4);
                y1=y1+(h/6)*(l1+2*(l2+l3)+l4);
                x0=x0+h;
        }
        printf("\n %f\t%f", x0, y0);
        return y0;
}

int main()
{
        float x0, yx0, yx1, xp, yxp, h;
        char q;
        printf("\n Enter the initial point x: ");
        scanf("%f", &x0);
        printf("\n Enter the value of y(x): ");
        scanf("%f", &yx0);
        printf("\n Enter the value of y'(x): ");
        scanf("%f", &yx1);
        printf("\n Enter the step length: ");
        scanf("%f", &h);

        do{
                printf("\n\n Enter the point x at which y(x) is required: ");
                scanf("%f", &xp);
                yxp=heun(x0, xp, yx0, yx1, h);
                printf("\n The approximate value of y(%f) is %f.", xp, yxp);
                printf("\n\n Do you want to approximate at another point?(y/n): ");
                scanf(" %c", &q);
        } while (q=='y');
        getch();
        return 0;
}
```

```
 /* Solving Boundary Value Problem using Shooting Method */
#include <stdio.h>
#include <math.h>
#include <conio.h>
#define EPS 0.00001
float f1(float x, float y1, float y2){
        return y2;
}
float f2(float x, float y1, float y2){
        return 2*x*x*y1+1;
}
//Routine for Euler's method
float euler(float x0, float x1, float y0, float y1, float h){
        float m1, m2;
        printf("\n\nCalculation of y(%f) for guess value %f:", x1, y1);
        printf("\n x \t\t y(x)");
        while (fabs(x0-x1)>0.0001){
                printf("\n %f\t%f", x0, y0);
                m1=f1(x0, y0, y1);
                m2=f2(x0, y0, y1);
                y0=y0+h*m1;
                y1=y1+h*m2;
                x0=x0+h;
        }
        printf("\n %f\t%f", x0, y0);
        return y0;
}
//Routine for Heun's method
float heun(float x0, float x1, float y0, float y1, float h){
        float m11, m12, m21, m22;
        printf("\n\nCalculation of y(%f) for guess value %f:", x1, y1);
        printf("\n x \t\t y(x)");
        while (fabs(x0-x1)>0.0001){
                printf("\n %f\t%f", x0, y0);
                m11=f1(x0, y0, y1);
                m21=f2(x0, y0, y1);
                m12=f1(x0+h, y0+h*m11, y1+h*m21);
                m22=f2(x0+h, y0+h*m11, y1+h*m21);
                y0=y0+0.5*h*(m11+m12);
                y1=y1+0.5*h*(m21+m22);
                x0=x0+h;
        }
        printf("\n %f\t%f", x0, y0);
        return y0;
}
//Routine for Runge-Kutta method
float rk(float x0, float x1, float y0, float y1, float h){
        float k1, k2, k3, k4, l1, l2, l3, l4;
        printf("\n\nCalculation of y(%f) for guess value %f:", x1, y1);
        printf("\n x \t\t y(x)");
        while (fabs(x0-x1)>0.0001){
                printf("\n %f\t%f", x0, y0);
                k1=f1(x0, y0, y1);
                l1=f2(x0, y0, y1);
                k2=f1(x0+0.5*h, y0+0.5*h*k1, y1+0.5*h*l1);
                l2=f2(x0+0.5*h, y0+0.5*h*k1, y1+0.5*h*l1);
                k3=f1(x0+0.5*h, y0+0.5*h*k2, y1+0.5*h*l2);
                l3=f2(x0+0.5*h, y0+0.5*h*k2, y1+0.5*h*l2);
```

```
            k4=f1(x0+h, y0+h*k3, y1+h*l3);
            l4=f2(x0+h, y0+h*k3, y1+h*l3);
            y0=y0+(h/6)*(k1+2*(k2+k3)+k4);
            y1=y1+(h/6)*(l1+2*(l2+l3)+l4);
            x0=x0+h;
        }
        printf("\n %f\t%f", x0, y0);
        return y0;
}
int main()
{
        float x0, yx0, x1, yx1, guess1, guess2, guess3, yguess1, yguess2, yguess3, xp, yxp, h;
        char q;
        printf("\n Enter the first boundary conditions x and y(x): ");
        scanf("%f%f", &x0, &yx0);
        printf("\n Enter the second boundary conditions x and y(x): ");
        scanf("%f%f", &x1, &yx1);
        printf("\n Enter the step length: ");
        scanf("%f", &h);
        printf("\n Enter the first guess of y'(%f): ", x0);
        scanf("%f", &guess1);
        yguess1=heun(x0, x1, yx0, guess1, h);
        printf("\n The calculated value of y(%f) is %f.", x1, yguess1);
        if (fabs(yguess1-yx1)<EPS) {
            guess3=guess1;
        }
        else {
        printf("\n\n Enter the second guess of y'(%f): ", x0);
        scanf("%f", &guess2);
        yguess2=heun(x0, x1, yx0, guess2, h);
        printf("\n The calculated value of y(%f) is %f.", x1, yguess2);
        if (fabs(yguess2-yx1)<EPS) {
            guess3=guess2;
            }
        else {
            do {
                guess3=guess2+(yx1-yguess2)*(guess1-guess2)/(yguess1-yguess2);
                yguess3=heun(x0, x1, yx0, guess3, h);
                guess1=guess2;
                guess2=guess3;
                yguess1=yguess2;
                yguess2=yguess3;
                } while (fabs(yguess2-yx1)>=EPS);
            printf("\n\n The extrapolated value of y'(%f) is %f.", x0, guess3);
            printf("\n\n The calculated value of y(%f) using y'(%f)=%f is %f.", x1, x0, guess3, yguess3);
            }
        }
        do{
            printf("\n\n Enter the point x at which y(x) is required: ");
            scanf("%f", &xp);
            yxp=heun(x0, xp, yx0, guess3, h);
            printf("\n The approximate value of y(%f) is %f.", xp, yxp);
            printf("\n\n Do you want to approximate at another point?(y/n): ");
            scanf(" %c", &q);
        } while (q=='y');
        getch();
        return 0;
}
```