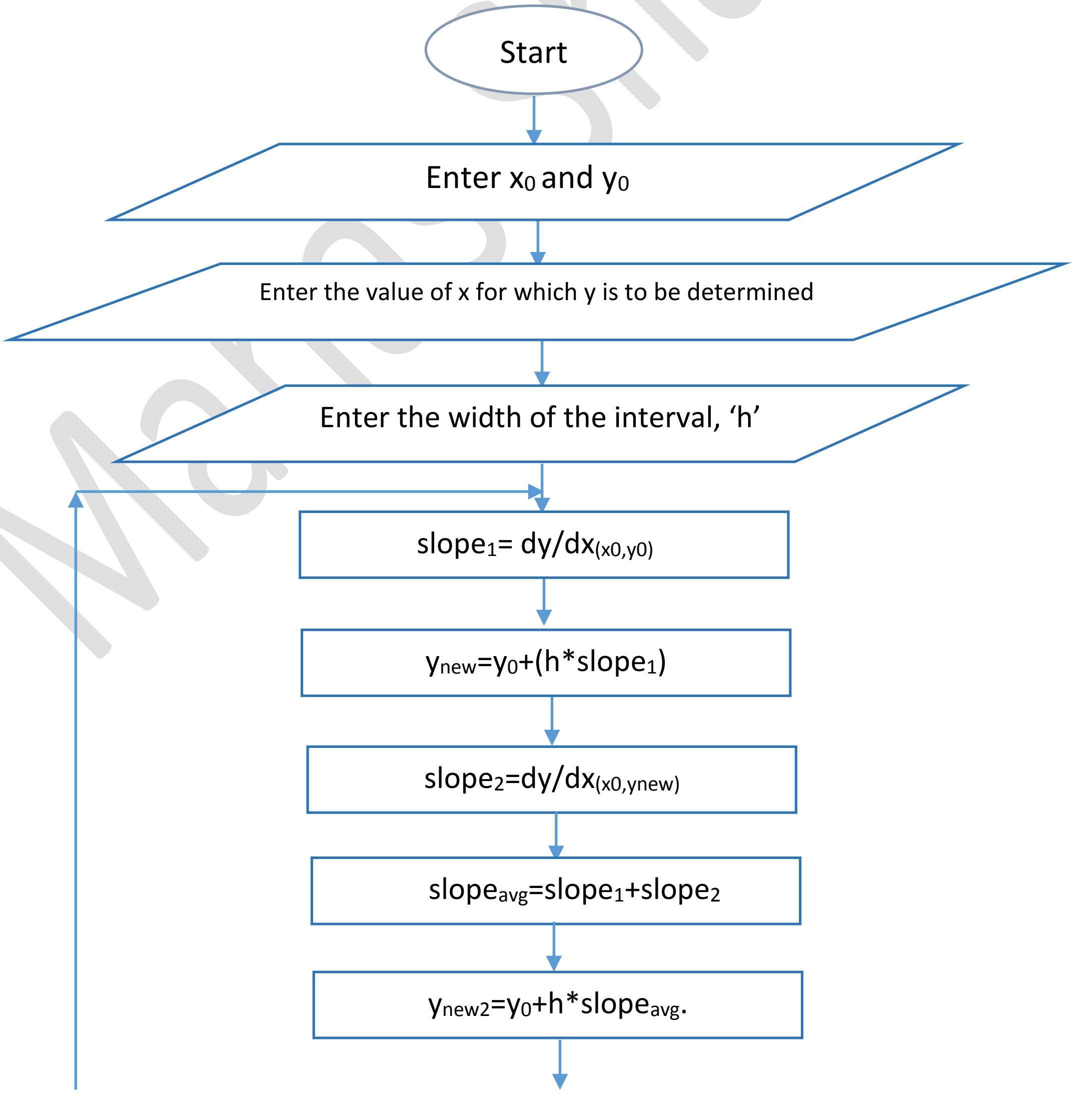
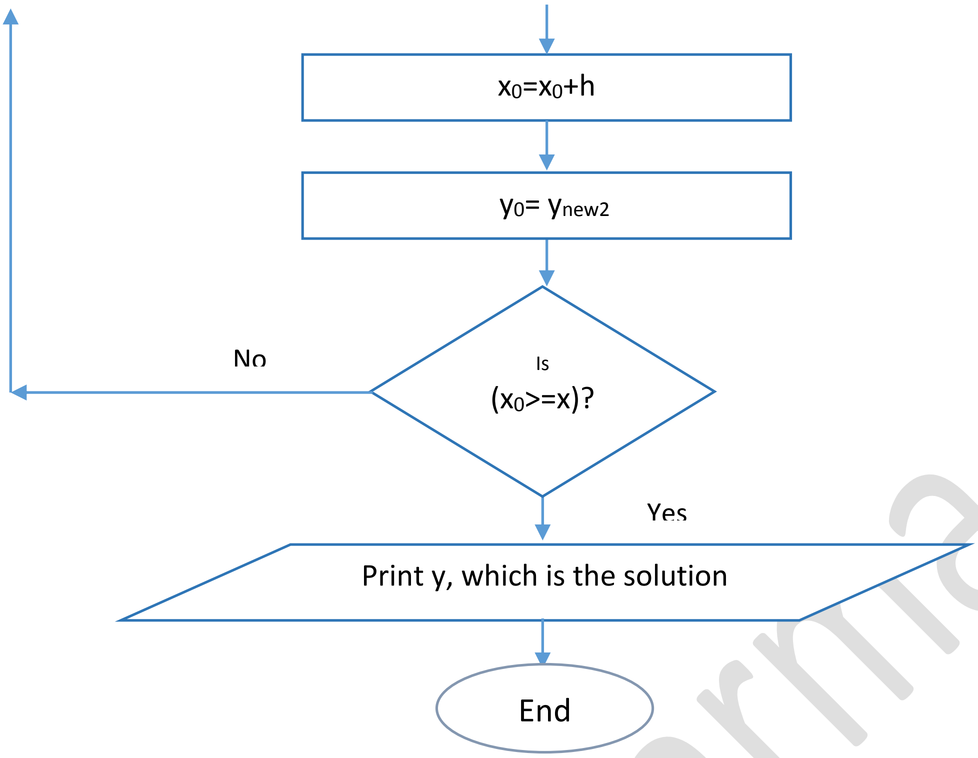
Aim: To solve a differential equation using modified Euler's Method.

Algorithm:

```
1. Enter the initial values of x and y($x_0$,$y_0$).
2. Enter the value of x for which y is to be determined.
3. Enter the width of the interval, 'h'.
4. Do:
        $slope_1$= $dy/dx_{(x0,y0)}$
        $y_{new}$=$y_0$+(h*$slope_1$)
        $slope_2$=$dy/dx_{(x0,ynew)}$
        $slope_{avg}$=$slope_1$+$slope_2$
        $y_{new2}$=$y_0$+h*$slope_{avg}$.
        $x_0$=$x_0$+h
        $y_0$= $y_{new2}$
   Until ($x_0$>=x)
5. Print y, which is the solution.
```

Flow Chart:


Start
Enter x_0 and y_0
Enter the value of x for which y is to be determined
Enter the width of the interval, 'h'
$slope_1$= $dy/dx_{(x0,y0)}$
y_{new}=y_0+(h*$slope_1$)
$slope_2$=$dy/dx_{(x0,ynew)}$
$slope_{avg}$=$slope_1$+$slope_2$
y_{new2}=y_0+h*$slope_{avg}$.



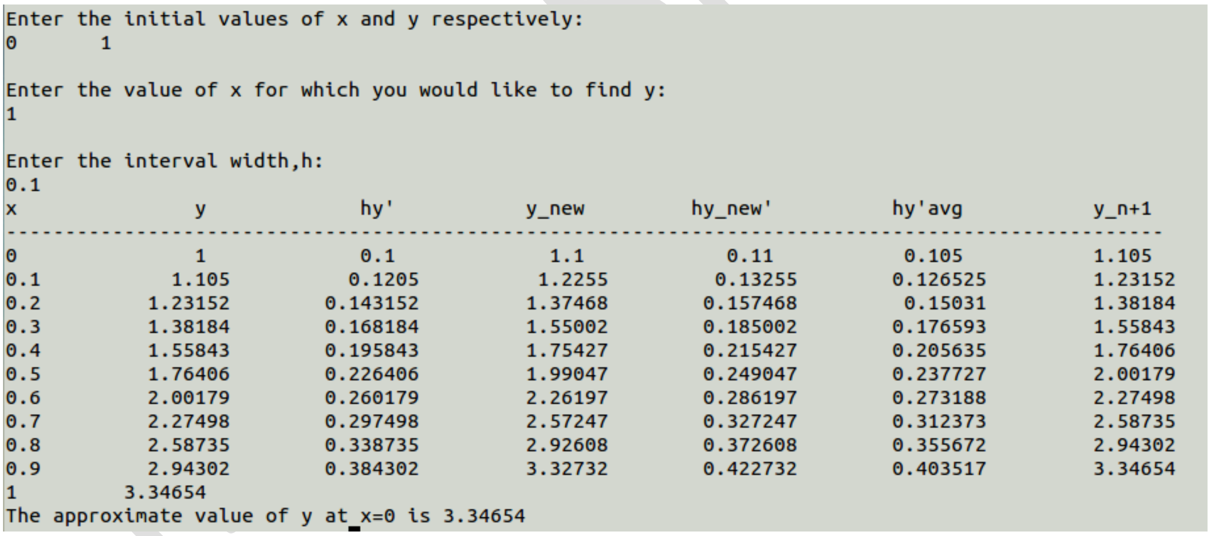
Program:

```
//Modified Euler's Method for differential equations
#include<iostream>
#include<iomanip>
#include<cmath>
using namespace std;
double df(double x, double y)
{
    double a=x+y;            //function for defining dy/dx
    return a;
}
int main()
{
    double x0,y0,x,y_i,dy1,dy2,dy_avg,y_n,h;    //for initial values, 
width, etc.
    cout<<"\nEnter the initial values of x and y respectively:\n";
    cin>>x0>>y0;                        //Initial values
    cout<<"\nEnter the value of x for which you would like to find y:\n";
    cin>>x;
    cout<<"\nEnter the interval width,h:\n";
    cin>>h;                            //input width

cout<<"x"<<setw(16)<<"y"<<setw(16)<<"hy'"<<setw(16)<<"y_new"<<setw(16)<<"hy
_new'"<<setw(16)<<"hy'avg"<<setw(16)<<"y_n+1"<<endl;
    cout<<"----------------------------------------------------------------
----------------------------------\n";
    while(fabs(x-x0)>0.0000001      //I couldn't just write "while(x0<x)" 
as they both are floating point nos. It is dangerous to compare two 
floating point nos. as they are not the same in binary as they are in 
decimal. For instance, a computer cannot exactly represent 0.1 or 0.7 in 
binary just like decimal can't represent 1/3 exactly without recurring 
digits.
```

```
    {
        dy1=h*df(x0,y0);         //calculate slope or dy/dx at x0,y0
        y_i=y0+dy1;             //calculate new y, which is y0+h*dy/dx
        dy2=h*df(x0,y_i);         //calculate slope or dy/dx at x0,new y
        dy_avg=(dy1+dy2)/2.0;         //calculate the average of the slopes
at y0 and new y
        y_n=y0+dy_avg;            //calculate new y, which is
y0+h*average(dy/dx)

cout<<x0<<setw(16)<<y0<<setw(16)<<dy1<<setw(16)<<y_i<<setw(16)<<dy2<<setw(1
6)<<dy_avg<<setw(16)<<y_n<<endl;
        x0=x0+h;            //calculate new x.
        y0=y_n;                //pass this new y as y0 in the next
iteration.
    }
    cout<<x0<<setw(16)<<y0<<endl;
    cout<<"The approximate value of y at x=0 is "<<y0<<endl;    //print the
solution.
    return 0;
}
```

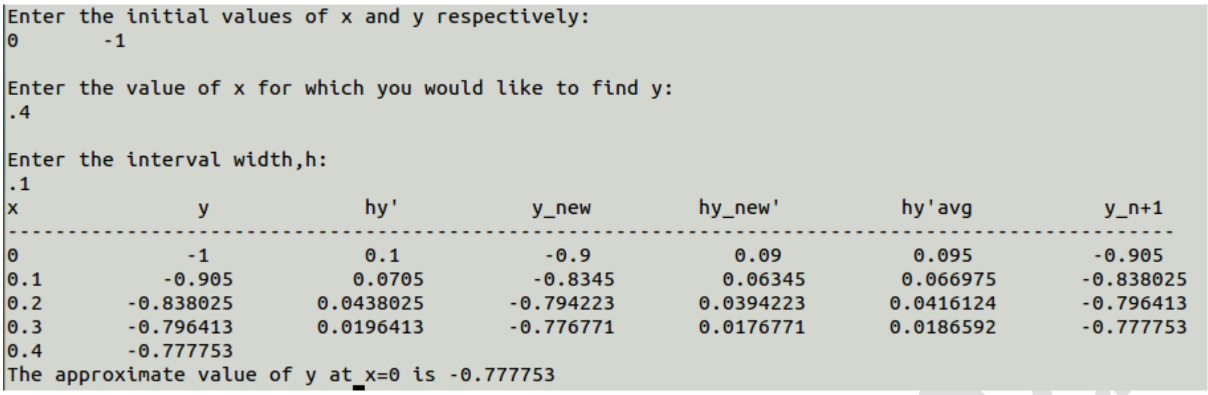
Outputs:

For dy/dx=x+y



```
Enter the initial values of x and y respectively:
0       1

Enter the value of x for which you would like to find y:
1

Enter the interval width,h:
0.1
x               y             hy'           y_new         hy_new'          hy'avg           y_n+1
-------------------------------------------------------------------------------------------------
0               1             0.1             1.1            0.11           0.105           1.105
0.1           1.105          0.1205          1.2255         0.13255        0.126525         1.23152
0.2         1.23152        0.143152         1.37468        0.157468         0.15031         1.38184
0.3         1.38184        0.168184         1.55002        0.185002        0.176593         1.55843
0.4         1.55843        0.195843         1.75427        0.215427        0.205635         1.76406
0.5         1.76406        0.226406         1.99047        0.249047        0.237727         2.00179
0.6         2.00179        0.260179         2.26197        0.286197        0.273188         2.27498
0.7         2.27498        0.297498         2.57247        0.327247        0.312373         2.58735
0.8         2.58735        0.338735         2.92608        0.372608        0.355672         2.94302
0.9         2.94302        0.384302         3.32732        0.422732        0.403517         3.34654
1         3.34654
The approximate value of y at x=0 is 3.34654
```

For dy/dx=-2x-y



```
Enter the initial values of x and y respectively:
0       -1

Enter the value of x for which you would like to find y:
.4

Enter the interval width,h:
.1
x               y             hy'           y_new           hy_new'           hy'avg           y_n+1
--------------------------------------------------------------------------------------------------
0              -1             0.1            -0.9            0.09            0.095          -0.905
0.1          -0.905          0.0705         -0.8345         0.06345         0.066975       -0.838025
0.2       -0.838025       0.0438025       -0.794223       0.0394223       0.0416124       -0.796413
0.3       -0.796413       0.0196413       -0.776771       0.0176771       0.0186592       -0.777753
0.4       -0.777753
The approximate value of y at x=0 is -0.777753
```