/\*integration of given function using Simpson's 1/3 rule of I=\*/

#include<stdio.h>

float y(float x)

{

return 1/(1+x\*x);

}

int main(){

float x0,xn,h,s;

int i,n;

printf("Enter x0, xn, no. of subintervals: ");

scanf("%f%f%d",&x0,&xn,&n);

h = (xn - x0)/n;

s = y(x0)+y(xn)+4\*y(x0+h);

for(i = 3; i<=n-1; i+=2){

s += 4\*y(x0+i\*h) + 2\*y(x0+(i-1)\*h);

}

printf("Value of integral is %6.4f\n",(h/3)\*s);

return0;

}

/\*Numerical solution of differential equation using Runge Kutta 4 order of dy/dx = \*/

#include<stdio.h>

#include <math.h>

#include<conio.h>

//dy/dx = 1 + y^2

#define F(x,y) 1 + (y)\*(y)

void main()

{

double y0,x0,y1,n,h,f,k1,k2,k3,k4;

clrscr();

printf("\nEnter the value of x0: ");

scanf("%lf",&x0);

printf("\nEnter the value of y0: ");

scanf("%lf",&y0);

printf("\nEnter the value of h: ");

scanf("%lf",&h);

printf("\nEnter the value of last point: ");

scanf("%lf",&n);

for(; x0<n; x0=x0+h)

{

f=F(x0,y0);

k1 = h \* f;

f = F(x0+h/2,y0+k1/2);

k2 = h \* f;

f = F(x0+h/2,y0+k2/2);

k3 = h \* f;

f = F(x0+h/2,y0+k2/2);

k4 = h \* f;

y1 = y0 + ( k1 + 2\*k2 + 2\*k3 + k4)/6;

printf("\n\n k1 = %.4lf ",k1);

printf("\n\n k2 = %.4lf ",k2);

printf("\n\n k3 = %.4lf ",k3);

printf("\n\n k4 = %.4lf ",k4);

printf("\n\n y(%.4lf) = %.3lf ",x0+h,y1);

y0=y1;

}

getch();

## } C Program to find the roots of Quadratic Equations

**/\*The following program finds out the roots of a quadriatic equation by taking the coefficents as the input from the user.\*/**

**#include<stdio.h>**

**#include<math.h>**

**int main()**

**{**

**double a,b,c;**

**printf("Enter the coefficents of the quadriatic equation of the form ax^2+bx+c\n");**

**scanf("%lf%lf%lf",&a,&b,&c);//We ask for the coefficents from the user**

**double D=pow(b,2)-4\*a\*c;**

**if(D<0)//We check the determinant whether it is greater or lesser than zero and accordingly find real or imaginary roots**

**{**

**printf("The roots are imaginary\n");**

**double real,img;**

**real=(-1\*b)/(2\*a);//The real part of the root**

**img=sqrt(-1\*D)/(2\*a);//The imaginary part of the root**

**printf("The first root is %lf+i%lf\n",real,img);**

**printf("The first root is %lf-i%lf\n",real,img);**

**}**

**if(D>=0)**

**{**

**printf("The roots are real\n");**

**if(D!=0)**

**{ //We find both roots if D>0**

**double root1=(-1\*b+sqrt(D))/(2\*a);**

**double root2=(-1\*b-sqrt(D))/(2\*a);**

**printf("The first root is %lf\n",root1);**

**printf("The second root is %lf\n",root2);**

**}**

**else**

**{ //We find the root if D==0 and display that we have equal roots**

**double root=(-1\*b)/(2\*a);**

**printf("The roots are equal\n");**

**printf("The root is %lf\n",root);**

**}**

**}**

**getch();**

**}**

**/\* Write a C program to generate GP series and find its sum.  
/\*General form of GP series is a, ar, ar^2, ar^3, ar^4, ar^5 ...\*/  
/\*a - First element of the sequence.\*/  
/\*r - common ratio.\*/  
/\*a = 2 and r = 5 GP series corresponds to these values are below.\*/  
/\*2, 10, 50, 250 ....\*/  
/\*Sum = a(1 - r^n)/(1 - r).  Where n is the number of elements in the series.\*/  
/\*Sum = 2(1 - 5^4)/ (1 - 5) = 312 <=> 2 + 10 + 50 + 250 \*/**

**#include <stdio.h>  
  #include <stdlib.h>  
  #include <math.h>  
#include<conio.h>  
void main()  
{  
        int \*data, value, ratio, sum, tmp, i, j, n;  
        /\* get the elements in series from user \*/  
        printf("Enter the number of elements:");  
        scanf("%d", &n);  
        /\* get first element of GP series from user \*/  
        printf("1st Element in Geomentric Sequence:");  
        scanf("%d", &value);  
        /\* get the ratio value from user \*/  
        printf("Enter the value for ratio:");  
        scanf("%d", &ratio);  
        tmp = value;  
        /\* allocate memory to store values in GP series \*/  
        data = (int \*)malloc(sizeof(int) \* n);  
        /\* stores the values of GP series in data array \*/  
        for (i = 0; i < n; i++) {  
                data[i] = tmp;  
                printf("%d ", data[i]);  
                tmp = value \* pow(ratio, i + 1);  
        }  
        /\* find the sum of the elements in GP series \*/  
        sum = (value \* (1 - pow(ratio, n)))/ (1 - ratio);**

**/\* print the results \*/**

**printf("\nSum of the elements in GP series is %d\n", sum);**

**getch ();**

**}**

**/\* SOLUTION Enter the number of elements:4\*/  
  /\*1st Element in Geomentric Sequence:2\*/  
/\*  Enter the value for ratio:5\*/  
 /\* 2 10 50 250 \*/  
  Sum of the elements in GP series is 312\*/   }**

**/\* Write a C program to sort numbers in ascending and descending order.\*/**

**#include <stdio.h>**

**#include <conio.h>  
void main()**

**{  
        int n, data[100], i, j, temp;  
        /\* get the number of entries \*/  
        printf("Enter your input for n:");  
        scanf("%d", &n);  
        /\* get the input data \*/  
        for (i = 0; i < n; i++)  
                scanf("%d", &data[i]);  
        /\* sort the given data in ascending order \*/  
        for (i = 0; i < n-1; i++) {  
                for (j = i + 1; j < n; j++) {  
                        if (data[i] > data[j]) {  
                                temp = data[i];  
                                data[i] = data[j];  
                                data[j] = temp;  
                        }  
                }**

**}  
        /\* data in ascending order \*/  
        printf("Ascending Order:\n");  
        for (i = 0; i < n; i++)  
                printf("%d\n", data[i]);  
        /\* data in descending order \*/  
        printf("\nDescending Order:\n");  
        for (i = n-1; i >= 0; i--)  
                printf("%d\n", data[i]);  
getch ();**

**}**

**/\* SOLUTION Enter your input for n:5 :   300   100   500   200   400\*/  
 /\* Ascending Order:   100   200   300   400   500\*/  
  /\*Descending Order:   500   400   300   200   100 \*/**

**/\* C Program for 4th Order Runge Kutta Method (Solution of 1st Order Differential Equation \*/**

**/\* function dy/dx= x+y2 \*/**

**#inslude<stdio.h>**

**# include<math.h>**

**# include<coio.h>**

**#define F(x,y) x+y\*y**

**/\*Let dy/dx = F(x,y) be the differential function whose solution to be Evaluate\*/**

**void main()**

**{**

**double x0,y0,x,y1,h,f,k1,k2,k3,k4;**

**clrscr(); /\* CLEAR SCREEN COMMAND OPTIONAL\*/**

**printf(“\n enter initial value of x0”);**

**scanf(“%f”,&x0);**

**printf(“\n enter initial value of y0”);**

**scanf(“%f”,&y0);**

**printf(“\n enter step size h”);**

**/\* If Step size is not given then h=(XFinal –xInitial)/n, n= Number of Points\*/**

**scanf(“%f”,&h)**

**printf(“\n enter final value of x”);**

**scanf(“%f”,&x);**

**for(x0<x;x0=x0+h)**

**f=F(x0,y0);**

**k1=h\*f;**

**f=F(x0+h/2,y0+k1/2);**

**k2=h\*f;**

**f=F(x0+h/2,y0+k2/2);**

**k3=h\*f;**

**f=F(x0+h,y0+k3);**

**k4=h\*f;**

**y1=y0+(k1+2\*k2+2\*k3+k4)/6;**

**printf(“\n First Coefficient k1=%f”, k1);**

**printf(“\n Second Coefficient k2=%f”, k2);**

**printf(“\n Third Coefficient k3=%f”, k3);**

**printf(“\n Fourth Coefficient k4=%f”, k4);**

**printf(“\n Solution at y(%x) =%f”,x, y1);**

**y0=y1;**

**}**

**getch();**

**/\* C Program for 4th Order Runge Kutta Method (Solution of 1st Order Differential Equation \*/**

**/\* function dy/dx=(x-y)/(x+y)**

**/\*Let dy/dx = F(x,y) be the differential function whose solution to be Evaluat**

**#include<stdio.h>**

**#include<math.h>**

**#define F(x,y) x\*x+y\*y\*y**

**float f(float x,float y);**

**int main()**

**{**

**float x0,y0,m1,m2,m3,m4,m,y,x,h,xn;**

**/\* k1,k2,k2,k4 and k is replaced by m1,m2,m3,m4 ,m\*/**

**printf("Enter x0,y0,xn,h:"); /\* xn= Final value of x\*/**

**scanf("%f %f %f %f",&x0,&y0,&xn,&h);**

**x=x0;**

**y=y0;**

**printf("\n\nX\t\tY\n");**

**while(x<xn)**

**{**

**m1=f(x0,y0);**

**m2=f((x0+h/2.0),(y0+m1\*h/2.0));**

**m3=f((x0+h/2.0),(y0+m2\*h/2.0));**

**m4=f((x0+h),(y0+m3\*h));**

**m=((m1+2\*m2+2\*m3+m4)/6);**

**y=y+m\*h;**

**x=x+h;**

**printf("%f\t%f\n",x,y);**

**}**

**}**

**float f(float x,float y)**

**{**

**float m;**

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

**return m;**

**}**

/\* **Write a C program to generate AP series and find its sum.**  
AP series is a series of numbers in which the difference between the consecutive elements is same.  
  
**Example:** 1, 3, 5, 7, 9, 11, 13, 15, 17, 19  
  
Here, the common difference is 2.  
  
And the sum of the AP series can be found by using the below formula.  
  
Sum = n \* (a1 + an)/2  => (10 \* (1 + 19)) / 10 = 100  
  
n is the number of elements in the AP series.  
a1 is the first element in the AP series.  
an is the last element in the AP series.\*/

#include <stdio.h>  
  #include <stdlib.h>  
  
void main()  
/\* If You write  int main() then use return0\*/{  
        int i, n, \*data, diff, sum, value = 1;  
  
        /\* get the number of elements in AP series \*/  
        printf("Enter the value for number of elements:");  
        scanf("%d", &n);  
  
        /\* get the common difference from the user \*/  
        printf("Common difference for AP series:");  
        scanf("%d", &diff);  
  
        /\* allocate memory to store the elements in AP series \*/  
        data = (int \*)malloc(sizeof(int) \* n);  
  
        /\* print the series and store the AP series in data array \*/  
        printf("AP series: ");  
        for (i = 0; i < n; i++) {  
                printf("%d ", value);  
                data[i] = value;  
                value = value + diff;  
        }  
  
        /\* find the sum of the given AP series \*/  
        sum = (n \* (data[0] + data[n - 1]))/2;  
  
        /\* print the result \*/  
        printf("\nSum of the AP series is %d\n", sum);  
  
**getch (); /\* you can also write return 0;\*/**   }

Enter the value for number of elements:10  
  Common difference for AP series:2  
  AP series: 1 3 5 7 9 11 13 15 17 19   
  Sum of the AP series is 100

/\* **Write a C program to generate GP series and find its sum.**  
General form of GP series is a, ar, ar^2, ar^3, ar^4, ar^5 ...  
a - First element of the sequence.  
r - common ratio.  
  
a = 2 and r = 5 GP series corresponds to these values are below.  
  
2, 10, 50, 250 ....  
  
Sum = a(1 - r^n)/(1 - r).  Where n is the number of elements in the series.  
Sum = 2(1 - 5^4)/ (1 - 5) = 312 <=> 2 + 10 + 50 + 250 \*/

#include <stdio.h>  
  #include <stdlib.h>  
  #include <math.h>  
  
void main()  
/\* If You write  int main() then use return0\*/

{  
        int \*data, value, ratio, sum, tmp, i, j, n;  
  
        /\* get the elements in series from user \*/  
        printf("Enter the number of elements:");  
        scanf("%d", &n);  
  
        /\* get first element of GP series from user \*/  
        printf("1st Element in Geomentric Sequence:");  
        scanf("%d", &value);  
  
        /\* get the ratio value from user \*/  
        printf("Enter the value for ratio:");  
        scanf("%d", &ratio);  
  
        tmp = value;  
  
        /\* allocate memory to store values in GP series \*/  
        data = (int \*)malloc(sizeof(int) \* n);  
  
        /\* stores the values of GP series in data array \*/  
        for (i = 0; i < n; i++) {  
                data[i] = tmp;  
                printf("%d ", data[i]);  
                tmp = value \* pow(ratio, i + 1);  
        }  
  
        /\* find the sum of the elements in GP series \*/  
        sum = (value \* (1 - pow(ratio, n)))/ (1 - ratio);

        /\* print the results \*/

        printf("\nSum of the elements in GP series is %d\n", sum);

**getch (); /\* you can also write return 0;\*/**   }

Enter the number of elements:4  
  1st Element in Geomentric Sequence:2  
  Enter the value for ratio:5  
  2 10 50 250   
  Sum of the elements in GP series is 312