Code

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

int main(){
int n,a=0,b=1,c,i;
clrscr();
printf(                    );
scanf(      ,&n);
printf(                  ,a,b);

for(i=1;i<=n;i++){
c=a+b;
a=b;
b=c;
printf(      ,b);
}

printf(                                  );
getch();
}
```

1:1

F1 Help F2 Save F3 Open Alt-F9 Compile F9 Make F10 Menu

Output

```
Enter the range: 10
0
1
1
2
3
5
8
13
21
34
55
89
Sukhmani Kaur CSE-1 00413202722_
```

Expt. No. 1

Practical No - 1

Q. Write a program to print Fibonacci Series upto a range of numbers.

- Description →

The Fibonacci series is a series in which each number is the sum of two preceding ones. Numbers that are part of fibonacci series are known as Fibonacci numbers, commonly denoted by F_n. The sequence starts from 0 and 1, the first few values in the sequence are :-

0, 1, 1, 2, 3, 5, 8, 13, 21, 34, ...

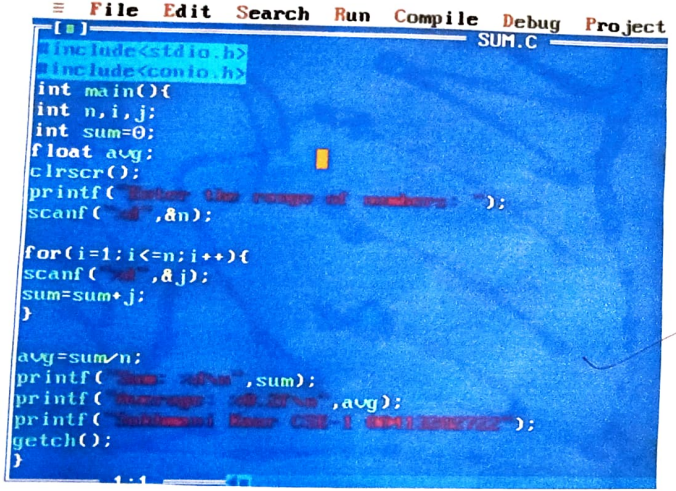
Thus, fibonacci numbers may be defined by the recurrence relation:

$$F_0 = 0 \quad , \quad F_1 = 1$$

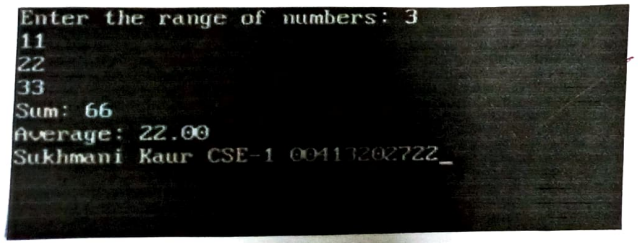
And $F_n = F_{n-1} + F_{n-2}$; $n \geq 1$

Teacher's Signature :

Code



Output



Expt. No. 2

Page No. 3

Practical No-2

Q. Write a program to find sum and average of the numbers given by user.

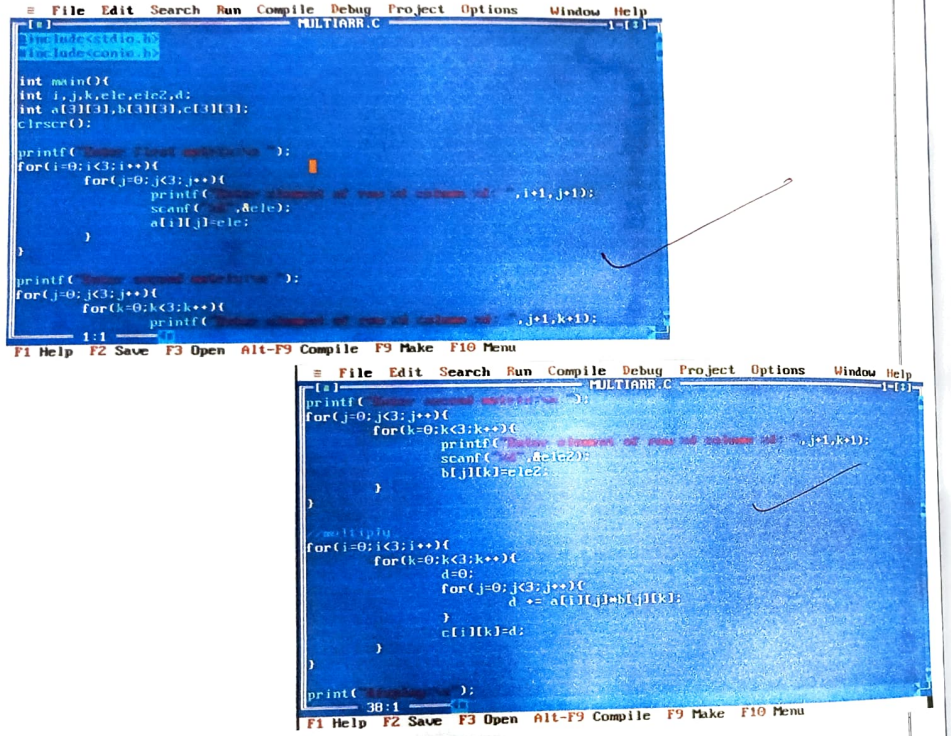
- **Description**

This program allows user to enter the numbers of which the sum and average are to be calculated.

Then with the help of the loop, the program will calculate the sum and average of the given numbers and provide the required output.

Teacher's Signature :

Code



Expt. No. 3

Page No. 4

Pratical No-3

Q Write a program to multiply two matrices (3X3) that are given/input by the user.

- **Description** –

Matrix = A rectangular array of $m \times n$ numbers where m is no. of rows and n is no. of columns.

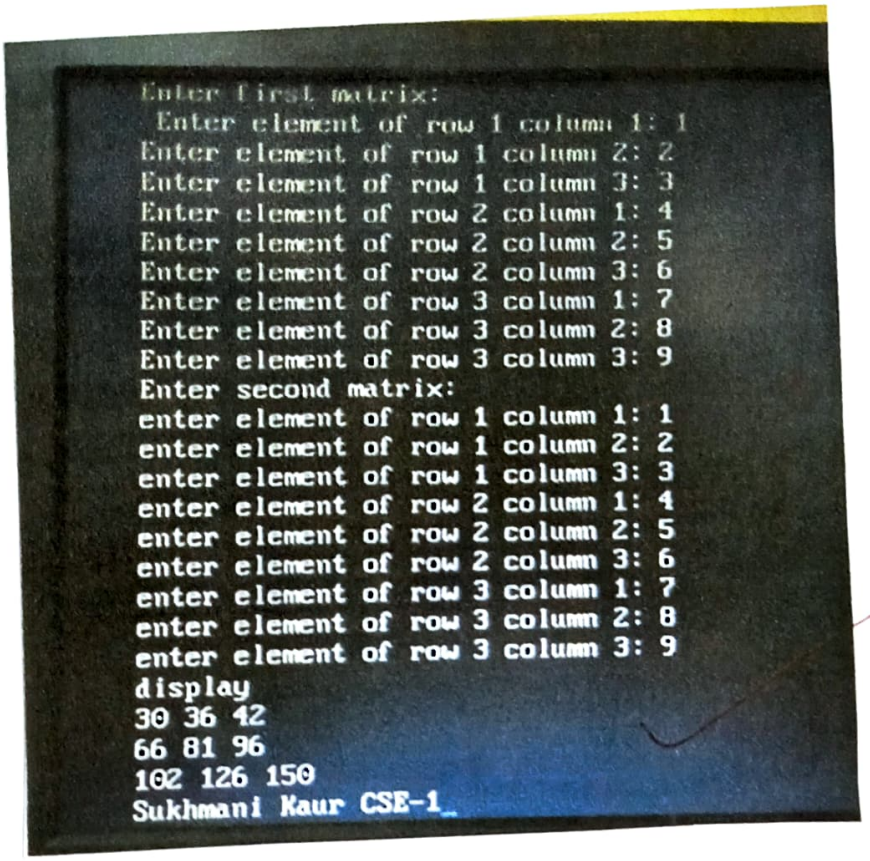
- Two matrices can be multiplied only if the number of columns in first matrix is equal to no. of rows in second matrix.

$$A_{ij} \times B_{jk} = C_{ik}$$

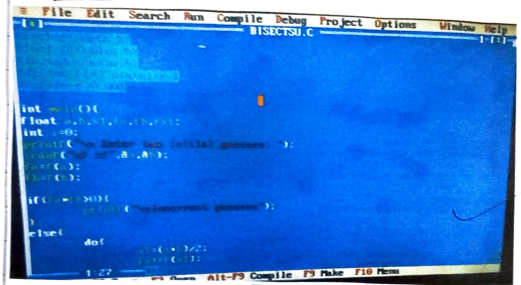
- The product of two matrices, A and B, is the sum of the products across some row of A with corresponding entries down some column of B.

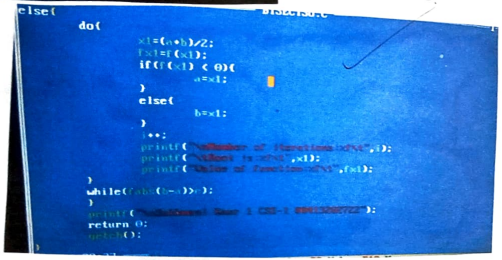
Teacher's Signature : _____

Output


Enter first matrix:
Enter element of row 1 column 1: 1
Enter element of row 1 column 2: 2
Enter element of row 1 column 3: 3
Enter element of row 2 column 1: 4
Enter element of row 2 column 2: 5
Enter element of row 2 column 3: 6
Enter element of row 3 column 1: 7
Enter element of row 3 column 2: 8
Enter element of row 3 column 3: 9
Enter second matrix:
enter element of row 1 column 1: 1
enter element of row 1 column 2: 2
enter element of row 1 column 3: 3
enter element of row 2 column 1: 4
enter element of row 2 column 2: 5
enter element of row 2 column 3: 6
enter element of row 3 column 1: 7
enter element of row 3 column 2: 8
enter element of row 3 column 3: 9
display
30 36 42
66 81 96
102 126 150
Sukhmani Kaur CSE-1

Code





Output

```
Enter two initial guesses: 1 2

Incorrect guesses
Sukhmani Kaur 1 CSE-1 00413202722
 Enter two initial guesses:
```

Date 27/9/23

Expt. No. 4 Page No. 6

Practical No-4

- **Aim** – Write a program to solve a given function using Bisection method.
- **Description** →

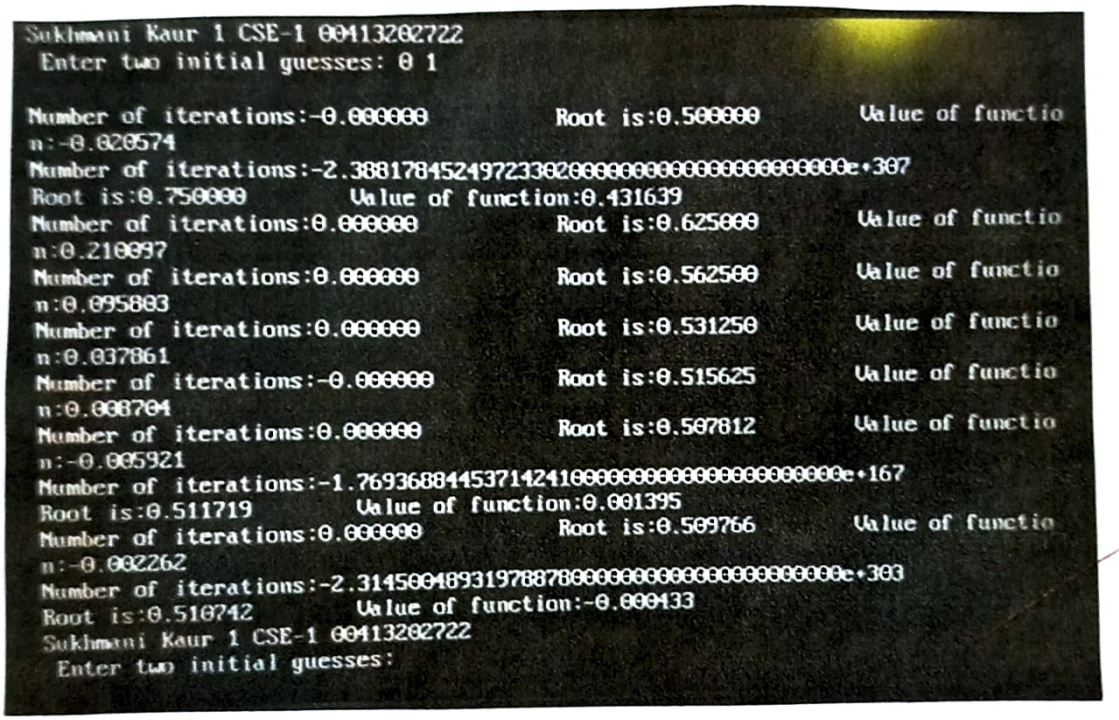
The method is also called interval halving method. This method is used to find root of an equation in a given interval that is value of 'x' for which $f(x)=0$.

It is based on Intermediate Value Theorem which states that if $f(x)$ is a continuous function and there are two real numbers such that $f(a)*f(b)<0$ then it is guaranteed to have atleast one root between them.

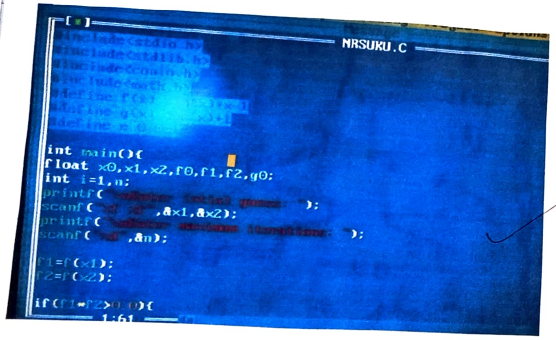
Steps –

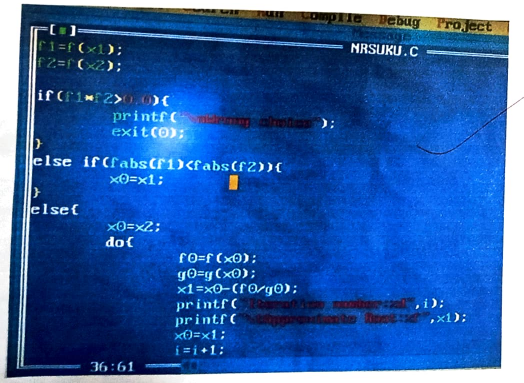
1. Let $f(x)$ be a continuous function in $[a,b]$, & then $f(a)f(b)<0$ then $x_1 = \frac{a+b}{2}$.
2. If $f(x_1)=0$, $x_1 \rightarrow$ root of $f(x)$ otherwise root lies b/w a & x_1 or x_1 & b as $f(x_1)>0$ or $f(x_1)<0$.

Teacher's Signature :


Sukhmani Kaur 1 CSE-1 00413202722
Enter two initial guesses: 0 1
Number of iterations:-0.000000 Root is:0.500000 Value of functio
n:-0.020574
Number of iterations:-2.388178452497233020000000000000000000000e+307
Root is:0.750000 Value of function:0.431639
Number of iterations:0.000000 Root is:0.625000 Value of functio
n:0.210097
Number of iterations:0.000000 Root is:0.562500 Value of functio
n:0.095883
Number of iterations:0.000000 Root is:0.531250 Value of functio
n:0.037861
Number of iterations:-0.000000 Root is:0.515625 Value of functio
n:0.008704
Number of iterations:0.000000 Root is:0.507812 Value of functio
n:-0.005921
Number of iterations:-1.769368844537142410000000000000000000000e+167
Root is:0.511719 Value of function:0.001395
Number of iterations:0.000000 Root is:0.509766 Value of functio
n:-0.002262
Number of iterations:-2.314500489319788780000000000000000000000e+303
Root is:0.510742 Value of function:-0.000433
Sukhmani Kaur 1 CSE-1 00413202722
Enter two initial guesses:

Code





Expt. No. 5 Date 11/10/23

Page No. 8

Practical No-5

- **Aim** — Write a program to solve a given function using Newton Raphson Method.

- **Description** →

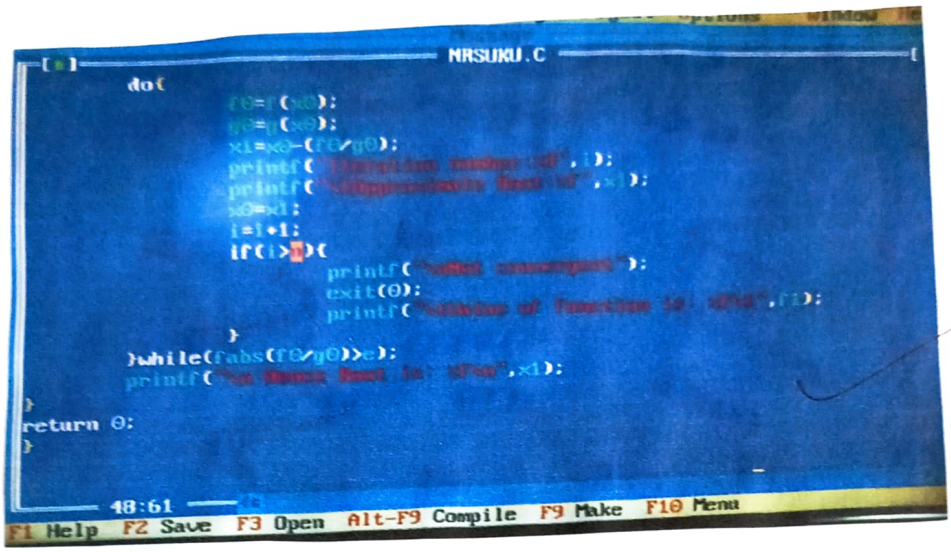
The Newton Raphson method uses below formula to find roots of the given equation.

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

- It is the best method to solve the non-linear equations.
- The order of convergence is quadratic i.e. it makes the method comparitively faster.

1. Compute value of $f(x)$ and derivative of $f(x)$ (i.e. $g(x)$)
2. Then put the values in the formula.

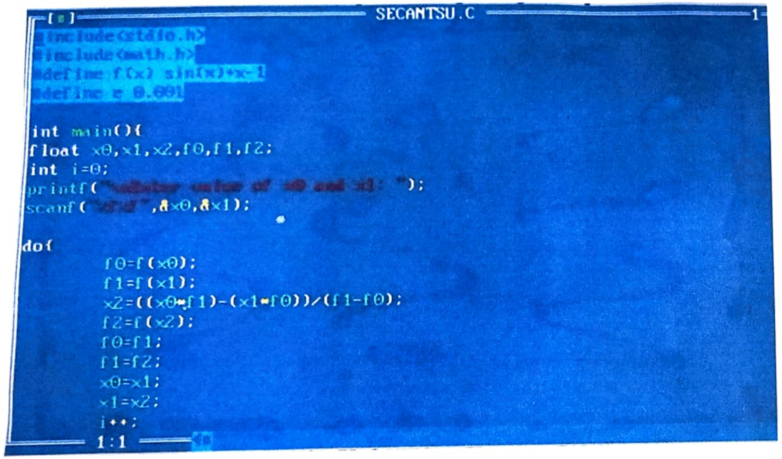
Teacher's Signature :

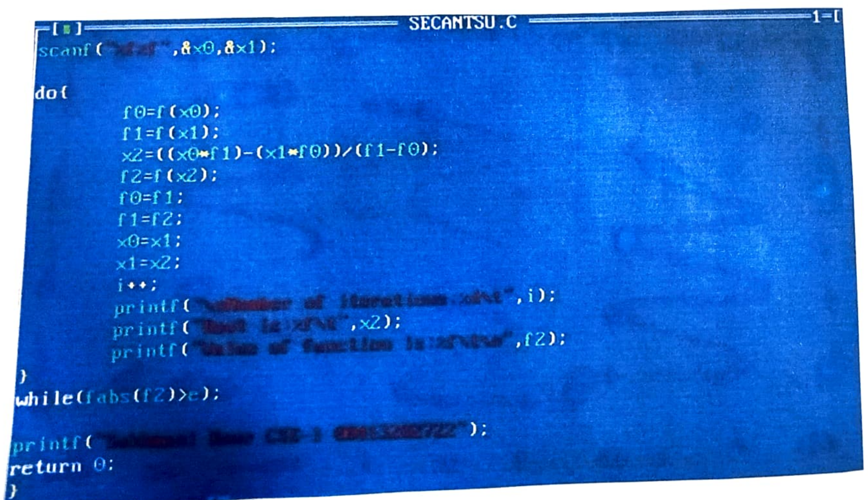


#Output

```
Enter intial guess: 0 1

Enter maximum iterations: 6

Iteration number:1      Approximate Root:0.453698
Iteration number:2      Approximate Root:0.510580
Iteration number:3      Approximate Root:0.510973
 Hence Root is: 0.510973
Sukhmani Kaur CSE-1 00413202722
Enter intial guess: 1 2

Enter maximum iterations: 2

Wrong choice
Enter intial guess: 0 1

Enter maximum iterations: 2

Iteration number:1      Approximate Root:0.453698
Iteration number:2      Approximate Root:0.510580
Not convergent
Enter intial guess:
```

Code





Expt. No. 6 Page No. 10

Date

Program - 6

- **Aim** — Write a program to find the number of iterations and root of solution of a given function by using secant method.

- **Description** —

The secant method is a root finding procedure in numerical analysis that uses a series of roots of secant lines to better approximate a root of a function.

- **Steps** —

Step1 : Initialization

x_0 and x_1 of α are taken as initial guess.

Step2: Iterations

In the case of $n = 1,2,3\ldots$

$$x_{n+1} = x_n - f(x_n) \cdot \frac{x_n - x_{n-1}}{f(x_n) - f(x_{n-1})}$$

until a specific criterion of termination has been met (i.e. the desired accuracy of the answer or the maximum no. of iterations has been attained.

Convergence

If the initials values x_0 & x_1 are close enough to the root, the secant method iterations x_n and converges to a root of

Teacher's Signature :

Output

```
Sukhmani Kaur CSE-1 00413202722
Enter value of x0 and x1: 0 1

Number of iterations:1  Root is:0.543044        Value of function is:0.059789

Number of iterations:2  Root is:0.508093        Value of function is:-0.005395

Number of iterations:3  Root is:0.510986        Value of function is:0.000023

Sukhmani Kaur CSE-1 00413202722
Enter value of x0 and x1: _
```

```
Sukhmani Kaur CSE-1 00413202722
Enter value of x0 and x1: 1 2

Number of iterations:1  Root is:0.211978        Value of function is:-0.577628

Number of iterations:2  Root is:0.627275        Value of function is:0.214215

Number of iterations:3  Root is:0.514926        Value of function is:0.007396

Number of iterations:4  Root is:0.510988        Value of function is:-0.000122

Sukhmani Kaur CSE-1 00413202722
Enter value of x0 and x1: _
```

Expt. No. ____ Date ____

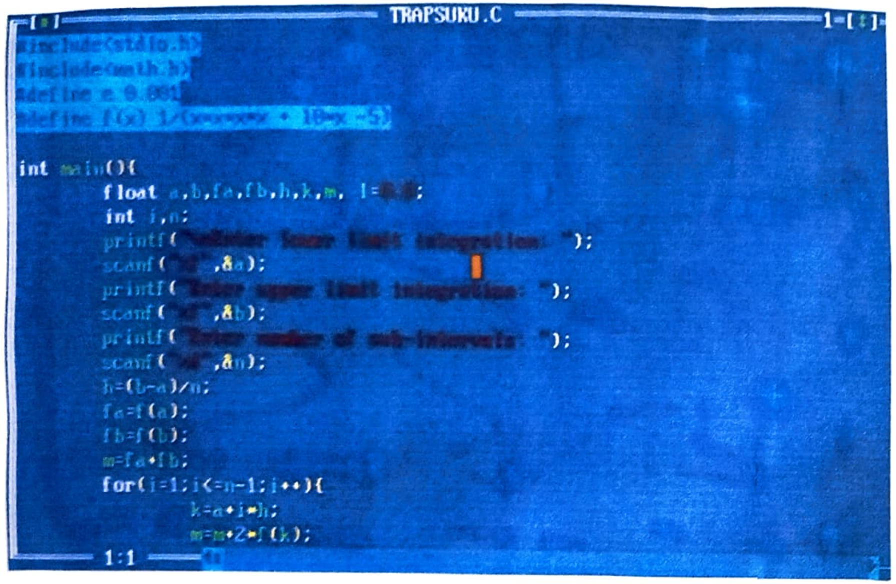
Page No. 11

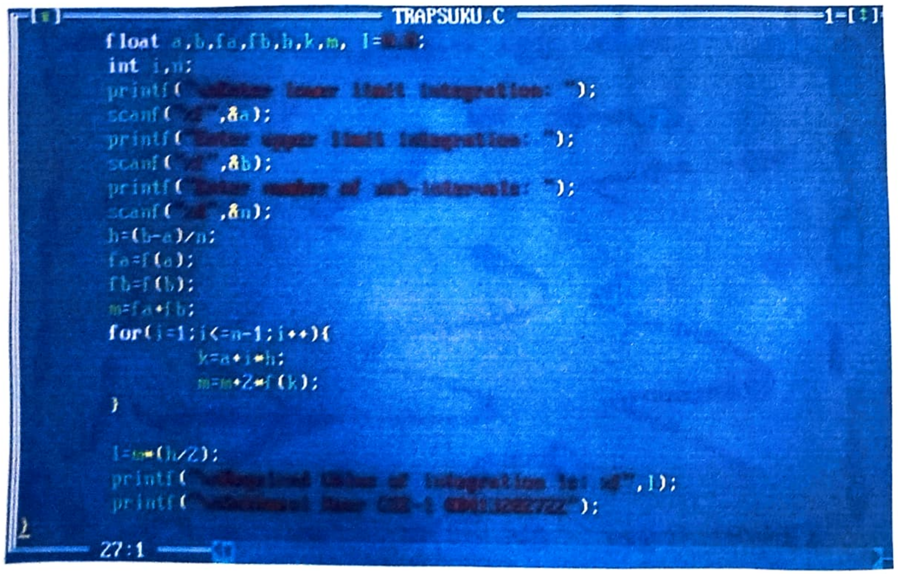
function f. The order of convergence is given by ϕ, where $\psi = \frac{1+\sqrt{5}}{2} \approx 1.618$ which is the Golden Ratio.

- **Advantages**
  - It converges quicker than a linear rate, making it more convergent than the bisection method.
  - It does not necessitate the usage of the function's derivative which is not available in number of application.
- **Disadvantages**
  - The secant method may not converge.
  - The computed iterates have no guaranteed error bounds.

Teacher's Signature : _____

Code





Program-7

- **Aim** — Write a program to find integral of function using Trapezoidal Rule.

- **Description** —

In calculus, "Trapezoidal Rule" is one of the most important integration rules. The name trapezoidal is known when the area under the curve is evaluated, then the total area is divided into small trapezoids instead of rectangles. This rule is used for approximately the definite integrals where it uses the linear approximation of the function. The trapezoidal rule is used mostly in the numerical analysis process. To evaluate the definite integrals, we can also use Riemann Sums, where we use small rectangles to evaluate the area under the curve.

Trapezoidal Rule is a rule that evaluates the area under the curves by dividing the total area into smaller trapezoids rather than using rectangles. This integration works by approximately the region under the graph of a function as a trapezoid and it calculates the area. This rule takes the average of the left and right sum. The trapezoidal rule does not give accurate value as simphsons rule uses the quadratic approximation instead of linear approximation. Both simpson's Rule and Trapezoidal Rule give the approximation value, but for simpsons rule results in even more accurate approximation value of the integrals.

Teacher's Signature : ____________

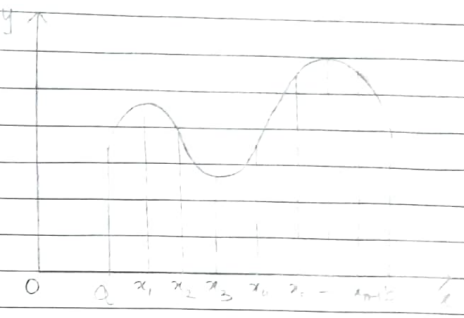
Output

```
Enter lower limit integration: 1
Enter upper limit integration: 2
Enter number of sub-intervals: 7

Required Value of integration is: 0.076863
Sukhmani Kaur CSE-1 00413202722
```

Date

Expt. No. .....

Page No. 13



Let $f(x)$ be a continuous function on the integral $[a,b]$. Now divide the intervals $[a,b]$ into n equal subintervals with each of width $\Delta x = \frac{b-a}{n}$, such that $a = x_0 < x_1 < x_2 < \cdots < x_n = b$.

Then the trapezoidal rule formula for area approximating the definite integrals →

$$\int_a^b f(x)\,dx = T_n = \frac{\Delta x}{2}\left[f(x_0) + 2\left[f(x_1) + f(x_2) \cdots + 2f(x_{n-1})\right] + f(x_n)\right]$$

where $x_i = a + i\Delta x$

If $n \to \infty$, RHS of the expression approaches the definite integral $\int_a^b f(x)\,dx$.

Teacher's Signature : _____

Code

```
SIMPSUKU.C
#include<stdio.h>
#include<math.h>
#define e 0.001
#define f(x) 1/(exp(x) * log(x) * 5)

int main(){
        int i,n;
        float a,b,l,h,k,m,p=0.0,q=0.0;
        printf("Enter value of lower limit: ");
        scanf("%f",&a);
        printf("Enter value of upper limit: ");
        scanf("%f",&b);
        printf("Enter the number of sub-intervals: ");
        scanf("%d",&n);
        h=(b-a)/n;
        l=f(a) + f(b);
        for(i=1;i<=n-1;i++){
                k=a+(i*h);
                if(i%2 == 0){
                        p=p+(2*f(k));
                }
1:1
```

```
SIMPSUKU.C
        scanf("%f",&a);
        printf("Enter value of upper limit: ");
        scanf("%f",&b);
        printf("Enter the number of sub-intervals: ");
        scanf("%d",&n);
        h=(b-a)/n;
        l=f(a) + f(b);
        for(i=1;i<=n-1;i++){
                k=a+(i*h);
                if(i%2 == 0){
                        p=p+(2*f(k));
                }
                else{
                        q=q+(4*f(k));
                }
                m=(h/3)*(l+p+q);
        }
        printf("Required value of Integration: %f\n",m);
        printf("Submitted Hour CSE-1 2021202722");
        return 0;
}
30:1
```

Expt. No. 8 Date

Page No. 14

Program-8

- **Aim** — Write a program to find integral of function using Simpson's ⅓ rule.

- **Description** —

Simpson's rule is one of the numerical methods which is used to evaluate the definite integral. Usually, to find the definite integral, we use the fundamental theorem of calculus. where we have to apply the antiderivative technique of integration. However, sometimes, it isn't easy to find the antiderivative of an integral line in scientific experiments, where the function has to be determined from the observed readings. Therefore, numerical methods are used to approximate the integral in such conditions. Other numerical methods used are trapezoidal rule, midpoint rule, left or right approximation using Riemann sums. Here we will use Simpson's ⅓ rule formula.

$$\int_a^b f(x)dx \approx S_n = \frac{\Delta x}{3}\left[f(x_0) + 4f(x_1) + 2f(x_2) + 4f(x_3) + \cdots + 2f(x_{n-2}) + 4f(x_{n-1}) + f(x_n)\right]$$

where n is the even number $\Delta x = (b-a)/n$ and $x_i = a + i\Delta x$ If we have $f(x) = y$, which is equally spaced between $[a,b]$ and if $a = x_0$, $x_1 = x_0 + h$, $x_2 = x_0 + 2h \cdots x_n = x_0 + nh$, where h is the difference between $[a,b]$ and if $a = x_0$ or we can say that $y_0 = f(x_0)$, $y_1 = f(x_1)$, $y_2 = f(x_2) \cdots y_n = f(x_n)$ are the analogous values of y with each value of x.

Teacher's Signature : _____

Output

```
Enter value of lower limit: 1
Enter value of upper limit: 2
Enter the number of sub-intervals: 6
Required value of itegration: 0.076311
Sukhmani Kaur CSE-1 00413202722
```

Date

Expt. No. .....

Page No. 15

Formula

$$\int_a^b f(x)\,dx = \frac{h}{3}\left[(y_0+y_n) + 4(y_1+y_3+y_5+\dots+y_{n-1}) + 2(y_2+y_4+y_6+\dots+y_{n-2})\right]$$

$x_0 = a \quad x_1 = a+b \quad x_2 = b$

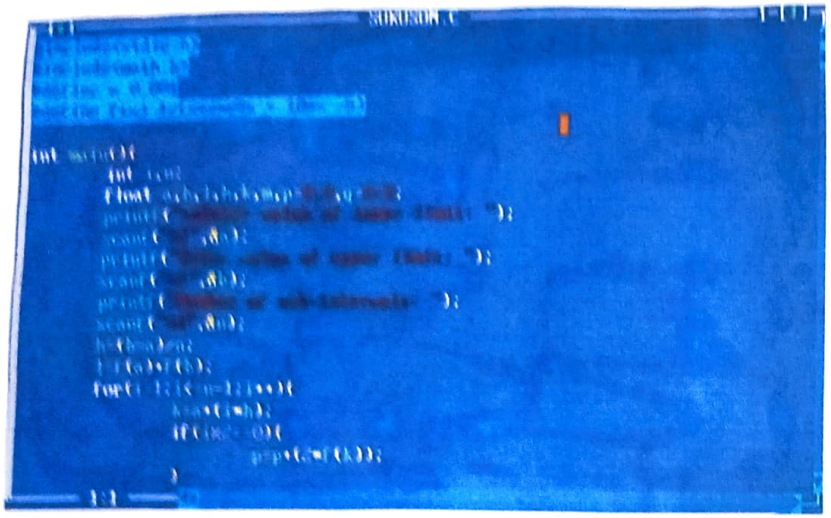
Hence,

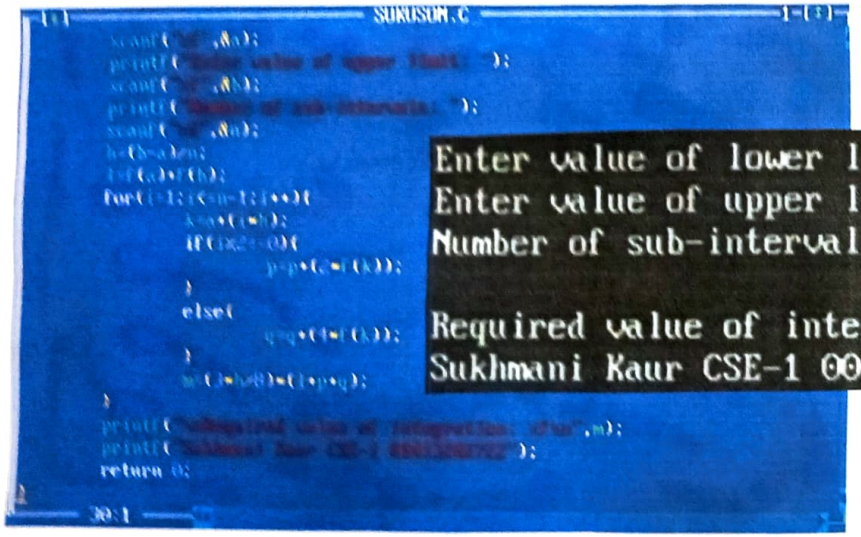
$$\int_a^b f(x)\,dx \simeq S_2 = \frac{h}{3}\left[f(x_0) + 4f(x_1) + f(x_2)\right]$$

$$S_2 = \frac{h}{3}\left[f(a) + 4f\left(\frac{a+b}{2}\right) + f(b)\right]$$

$h = \frac{b-a}{2}$, This is the Simpson's 1/3 Rule for integration.

Teacher's Signature : _____





Date

Expt. No. 9 Page No. 16

Program-9

- **Aim** – Write a program to find the integral of function using simpson's 3/8 Rule.

- **Description** –

Simpson's rule is one of the numerical methods which is used to evaluate the definite integral. Usually, to find the definite integral, we use the fundamental theorem of calculus where we have to apply the antiderivative techniques of integration. However, sometimes it isn't easy to find the antiderivative of an integral, like in scientific. Experiments, where the function has to be determined from the observed readings. Therefore, numerical methods are used to approximate the integral in such conditions. Other numerical methods used are trapezoidal rule, midpoint rule, left or right approximation using Rieman sums.

$$\frac{3h}{8}\left[(y_0+y_n)+3(y_1+y_2+y_4+y_5+\cdots+y_{n-1})+2(y_3+y_6+y_9+\cdots+y_{n-3})\right]$$

more accurate than the standard method, as it ore functional value. For 3/8 rule, the composite Simpsons 3/8 rule also exists which is similar to the generalized form. The 3/8 rule is known as Simpson's second Rule of integration.

Teacher's Signature : _____

#Code

LAGRANGE.C

```
void main(){
        float xp,x[20],y[20],yp=0.0,p;
        int i,j,n;
        printf("Enter the number of data: ");
        scanf("%d",&n);
        printf("Enter value to be found: ");
        scanf("%f",&xp);
        printf("Enter data:\n");
        for(i=0;i<n;i++){
                printf("x[%d]= ",i);
                scanf("%f",&x[i]);
                printf("y[%d]= ",i);
                scanf("%f",&y[i]);
        }

        for(i=0;i<n;i++){
                p=1;
                for(j=0;j<n;j++){
                        if(j!=i){
                                p=p*((xp-x[j])/(x[i]-x[j]));
                        }
                }
                yp=yp+p*y[i];
        }
        printf("Interpolated value at %.3f is %.3f\n",xp,yp);
        printf("Sukhmani Kaur CSE-1 00413202722");
        return 0;
}
```

1:19

#Output

```
Enter the number of data: 4
Enter value to be found: 10
Enter data:
x[0]= 5
y[0]= 12
x[1]= 6
y[1]= 13
x[2]= 34
y[2]= 56
x[3]= 99
y[3]= 36
Interpolated value at 10.000 is 17.565
Sukhmani Kaur CSE-1 00413202722
```

Date

Page No. 17

Program-10

– To find the value of function using lagrange's interpolation using 'C'.

Description–

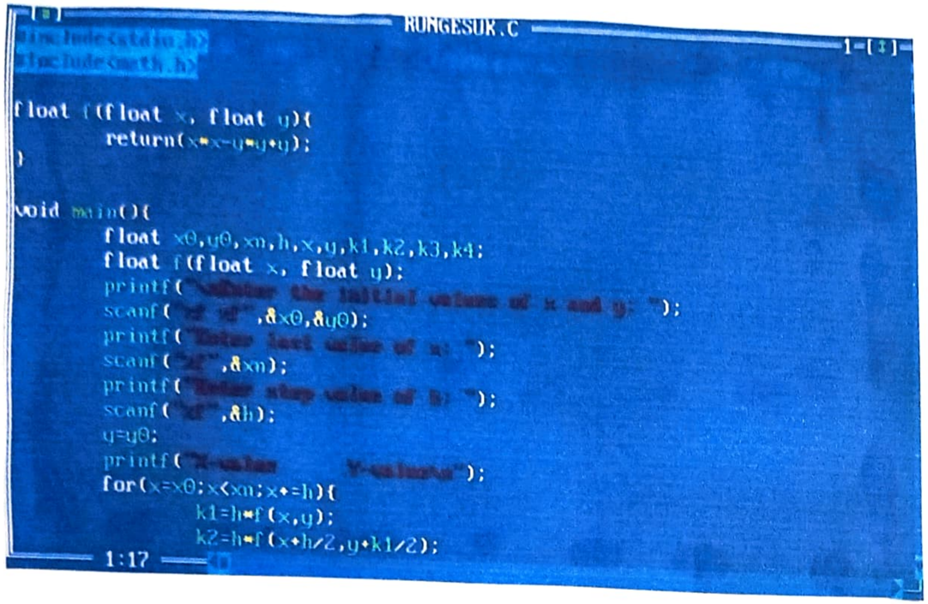
Lagrange Interpolation formula – find a polynomial called lagrange polynomial that takes on certain values at an arbitary point. It is an n^{th} degree polynomial expression to the function $f(x)$. The interpolation method is used to find the new data points within the range of a discrete set of known data points.

$$y = \frac{(x-x_1)(x-x_2)\cdots(x-x_n)}{(x_0-x_1)(x_0-x_2)\cdots(x_0-x_n)} \times y_0 + \frac{(x-x_0)(x-x_2)\cdots(x-x_n)}{(x_1-x_0)(x_1-x_2)\cdots(x_1-x_n)} y_1$$

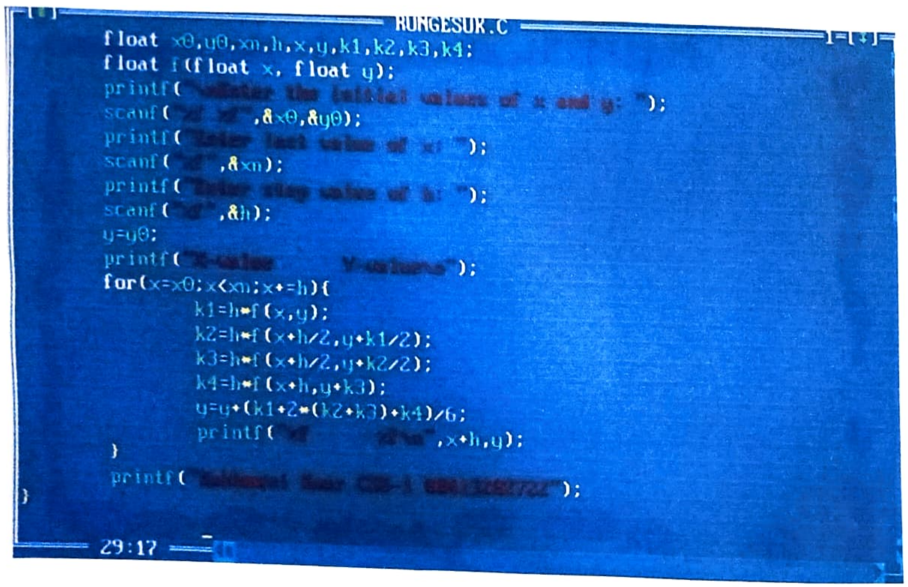
$$+ \cdots + \frac{(x-x_0)(x-x_1)\cdots(x-x_{n-1})}{(x_n-x)(x_0-x_1)\cdots(x_n-x_{n-1})} y_n$$

Teacher's Signature : _____

11 Code



```
RUNGESUR.C
#include<stdio.h>
#include<math.h>

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

void main(){
        float x0,y0,xn,h,x,y,k1,k2,k3,k4;
        float f(float x, float y);
        printf("\nEnter the initial values of x and y: ");
        scanf("%f %f",&x0,&y0);
        printf("Enter last value of x: ");
        scanf("%f",&xn);
        printf("Enter step value of h: ");
        scanf("%f",&h);
        y=y0;
        printf("X-value        Y-values\n");
        for(x=x0;x<xn;x+=h){
                k1=h*f(x,y);
                k2=h*f(x+h/2,y+k1/2);
1:17
```



```
RUNGESUR.C
        float x0,y0,xn,h,x,y,k1,k2,k3,k4;
        float f(float x, float y);
        printf("\nEnter the initial values of x and y: ");
        scanf("%f %f",&x0,&y0);
        printf("Enter last value of x: ");
        scanf("%f",&xn);
        printf("Enter step value of h: ");
        scanf("%f",&h);
        y=y0;
        printf("X-value        Y-values\n");
        for(x=x0;x<xn;x+=h){
                k1=h*f(x,y);
                k2=h*f(x+h/2,y+k1/2);
                k3=h*f(x+h/2,y+k2/2);
                k4=h*f(x+h,y+k3);
                y=y+(k1+2*(k2+k3)+k4)/6;
                printf("%f        %f\n",x+h,y);
        }
        printf("[illegible]");
}
29:17
```

Expt. No. 11 Date

Page No. 18

Program-11

- **Aim** — Write a program in C to solve the differential equation using Runge-Kutta method.

- **Description** —

Range Kutta Method — A method of numerically integrating ordinary differential equations by using a trial step at the mid point of an interval to cancel out lower order differential equation. Consider an ordinary differential equation of form $dy/dx = f(x,y)$ with initial condition $y(x_0) = y_0$. For this, we can define the formulas for Range-Kutta methods as follows :-

Ist order Runge-Kutta Method

$y_1 = y_0 + hf(x_0, y_0) = y_0 + hy_0'$ $\{$since $y' = f(x,y)\}$
same as euler's method.

2nd order runge-kutta method

$$y_1 = y_0 + \frac{1}{2}(k_1 + k_2)$$

$$k_1 = hf(x_0, y_0) \qquad k_2 = hf(x_0 + h, y_0 + k_1)$$

3rd order Runge-Kutta Method

$$y_1 = y_0 + \left(\frac{1}{6}\right)(k_1 + 4k_2 + k_3)$$

$\forall$ $k_1 = hf(x_0, y_0)$ $\quad k_2 = hf[x_0 + (1/2)h, \; y_0 + (1/2)k_1]$
$k_3 = hf[x_0 + h, y_0 + k_1]$ such that $k_1 = hf[x_0 + h, y_0 + k_1]$

Teacher's Signature :

#Output

```
Enter the initial values of x and y: 0 1
Enter last value of x: 0.5
Enter step value of h: 0.1
X-value         Y-value
0.100000        1.000325
0.200000        1.002538
0.300000        1.008361
0.400000        1.019339
0.500000        1.036845
Sukhmani Kaur CSE-1 00413202722
```

Date

Expt. No. .....

Page No. 19

4th order

The Runge kutta method provides the approximate value of y for a given point x. Only the first order ODE's can be solved using the Runge kutta Method.

$$y_1 = y_0 + \left(\frac{1}{6}\right)(k_1 + 2k_2 + 2k_3 + k_4)$$

$$k_1 = hf(x_0, y_0)$$
$$k_2 = hf\{x_0 + (1/2)h,\ y_0 + (1/2)k_1\}$$
$$k_3 = hf[x_0 + (1/2)h,\ y_0 + (1/2)k_2]$$
$$k_4 = hf[x_0 + h_1 y_0 + k_3]$$

Teacher's Signature : _____