Object-Oriented Programming Using C

[ES 202]

Practical file

AMITY SCHOOL OF ENGINEERING AND TECHNOLOGY

AMITY UNIVERSITY, UTTAR PRADESH

Submitted by: PRIYANSH PRATAP SINGH

A2305222395

1CSE6-Y

INDEX

S.No	Name Of Experiment	Date of experiment	Date of submission	Remarks
1A	Write a C program to add two numbers			
1B	Write a C program to add three numbers			
2A	Write a C program to find the area of the circle			
2B	Write a C program to find simple interest			
3	Write a C program to print a block F using hash (#), where the F has a height of six characters and width of five and four characters			
4	Write a C program that accepts two item's weights (floating points' values) and number of purchases (floating points' values) and calculate the average value of the items			
5A	Write a C program to swap two variables using a third variable			
5B	Write a C program to swap two variables without using a third variable			
6A	Write a C program to convert a given integer (in seconds) to hours, minutes, and seconds			
6B	Write a C program to convert specified days into years, weeks, and days			
6C	Write a C program to check whether a number is even or odd.			
7	Write a C program to check whether a given year is Leap year or not			
8A	Write a C program to check whether a triangle is Equilateral, scalene or isosceles			
8B	Write a C program to check whether a triangle is right angles,			
9	Write a C program to convert temperature from Fahrenheit to Celsius and Celsius to			
10A	Write a C program to check whether a character is an alphabet, digit			
10B	Write a C program a program to check whether an alphabet is a vowel or consonant			
11A	Write a C program a program to check whether an alphabet is a vowel or consonant			
11 B	Write a C program to find the largest of three numbers			
12	Write a program in C to implement Simple			
13	WAP to calculate the root of a Quadratic			
14	WAP to accept a coordinate point in a Xan coordinate system and determine in which quadrant the coordinate point lies.			
15	Write a program to find gross salary of employees if DA is 40% of basic Salary and HRA is 20% of basic salary. Basic salary will be entered as input by keyboard.			

16	Write a program in C to calculate and print the Electricity bill of a given customer. The customer id and unit consumed by the user should be taken from the keyboard and display the total amount to pay to the customer. upto 199 1.20 200-500----------1.80 Above 500 2.00 If bill exceeds Rs. 400 then a surcharge of 15% will be charged and the minimum bill should be of Rs. 100/-			
17	Write a program to accept the number of days the member is late to return the book and display the fine or appropriate message.			
18	Write a program to find the factorial of any number			
19	Write a program to print the Fibonacci sequence 0 1 1 2 3 5 8 13…… N terms and prints the sum of the sequence.			
20	Write a program in C to accept an integer number and find the sum			
21	Write a program in C to accept an integer number and find the reverse of this number and check this number for palindrome			
22	Write a program in C to accept an integer number and to check whether the number is Armstrong or not			
23	Write a program in C to accept an integer number and to check whether number is Perfect or not			
24	Write a program to find the sum of the following series: S = 2+4+6+8+… N terms.			
25	Write a program to check a number whether it is a prime number or not			
26	Write a program to find the sum of the following series: 1 – 1/2 + 1/3 – 1/4 + 1/5 - …… up to n terms.			
27	Write a program to find the sum of the following series: 1! + 2! + 3! + 4! +……..+ n!			
28	Write a program to find the sum of the following series: S = $-1^3 + 3^3 - 5^3 + 7^3 – 9^3 + 11^3$ N terms.			
29	Write a program to find the sum of the following series: S = 1/1! + 2/2! + 3/3! + 7 terms			
30	Write a program to convert binary numbers to decimal numbers			
31	Write a program to find the sum of the following series: S = $1^4 + 3^4 + 5^4 + 7^4 +$ 100 terms			
32	Write a program in C to print the following * * * * * * * * *			
33	Write a program in C to print the following pattern: 1 2 3 1 2 3 1 2 3			
34	Write a program in C to print the following pattern: 1 1 1 2 2 2 3 3 3			
35	Write a program in C to print the following pattern:			

	3 2 1 3 2 1 3 2 1			
36	Write a program in C to print the following pattern: 3 3 3 2 2 2 1 1 1			
37	Write a program in C to print the following * * * * * *			
38	Write a program in C to print the following pattern: 1 1 2 1 2 3			
39	Write a program in C to print the following pattern: 1 2 2 3 3 3			
40	Write a program in C to print the following pattern: 3 3 2 3 2 1			
41	Write a program in C to print the following pattern: 3 2 2 1 1 1			
42	Write a program in C to print the following pattern: * * * * * * * * * * * * * * * *			
43	Write a program in C to print the following pattern: 1 1 2 1 1 2 3 2 1 1 2 3 4 3 2 1			
44	Write a program in C to print the following pattern: 5 5 4 5 5 4 3 4 5 5 4 3 2 3 4 5 5 4 3 2 1 2 3 2 1			
45	Write a program in C to print the following pattern: 1 0 1 1 0 1 0 1 0 1 1 0 1 0 1			
46	Write a program to print all prime numbers <= a given number.			
47	Write a program to convert Decimal no to Binary No			

48	Write a program to find product, sum, average, max and min from a list of n numbers.			
49	Write a program in C to display the index of smallest and largest element in 10 integers			
50	Write a program in C to display the index of smallest and largest element in 3 X 4 matrix of integers.			
51	Write a program in C that accepts N*N matrix as input and print transpose of this matrix			
52	Write a program to accept two matrices of some order. (Order must be given by user) find out the sum of these matrices and print the sum of matrices			
53	Write a program to find out the product/ Multiplication of two matrices and print the product matrix			
54	Write a program to accept two matrices of some order. (Order must be given by user) find out the subtraction of these matrices and			
55	Write a C Program to implement Simple Calculator (Addition, Subtraction, Multiplication, Division) using the concept of function			
56	Write a C Program to swap two values using function			
57	Write a C Program to Calculate the factorial of a number using function			
58	Write a C Program to Calculate the factorial of a number using recursion			

1. (a) Write a C program to add two numbers.

Language used : C

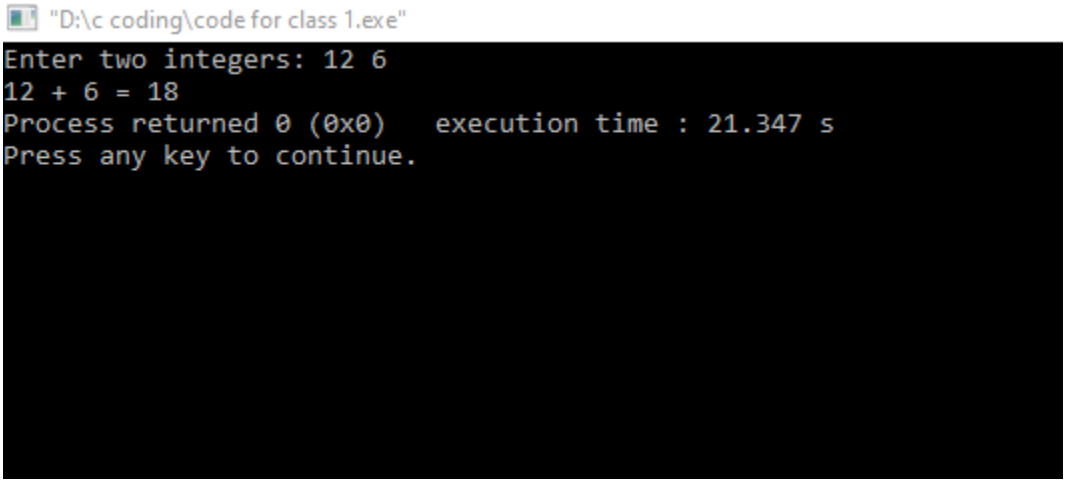
Theory : We have to print the sum of the two numbers entered by the user by using

sum = num1 + num2.

Code:

```
#include <stdio.h>

int main()

{

   int num1, num2, sum;

   printf("Enter two integers: ");

   scanf("%d %d", &num1, &num2);

   sum = num1 + num2; // calculating sum

   printf("%d + %d = %d", num1, num2, sum);

   return 0;

}
```

Output:



(b) Write a C program to add three numbers.

Language used : C

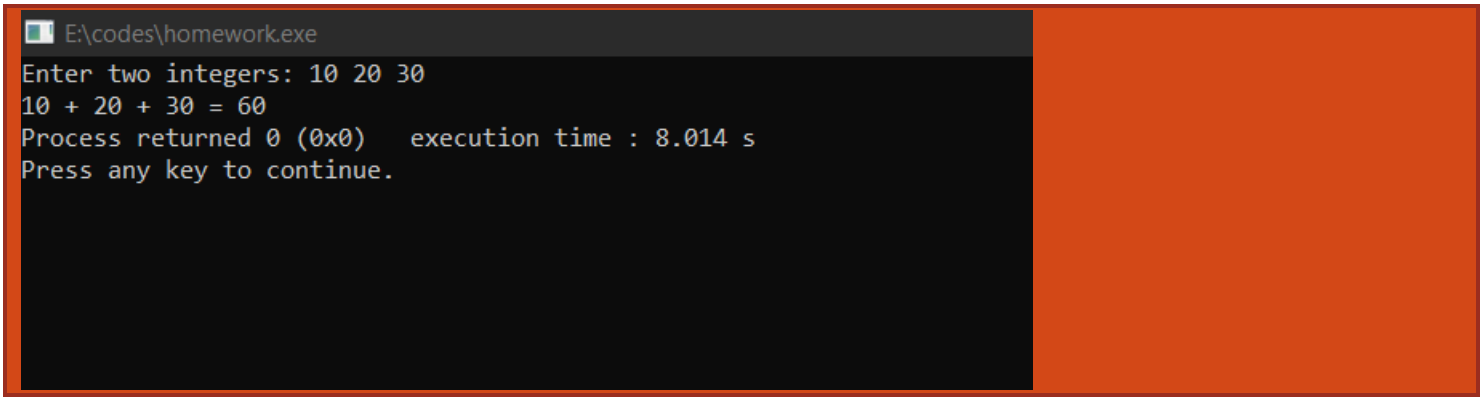
Theory : We have to print the sum of the three numbers entered by the user by using

sum = num1 + num2 + num3.

Code:

```
#include <stdio.h>
int main()
{
   int num1, num2, num3, sum;
   printf("Enter two integers: ");
   scanf("%d %d %d", &num1, &num2, &num3);
   sum = num1 + num2 + num3; // calculating sum
   printf("%d + %d + %d = %d", num1, num2, num3, sum);
   return 0;
}
```

Output:



2. (a) Write a C program to find area of circle.

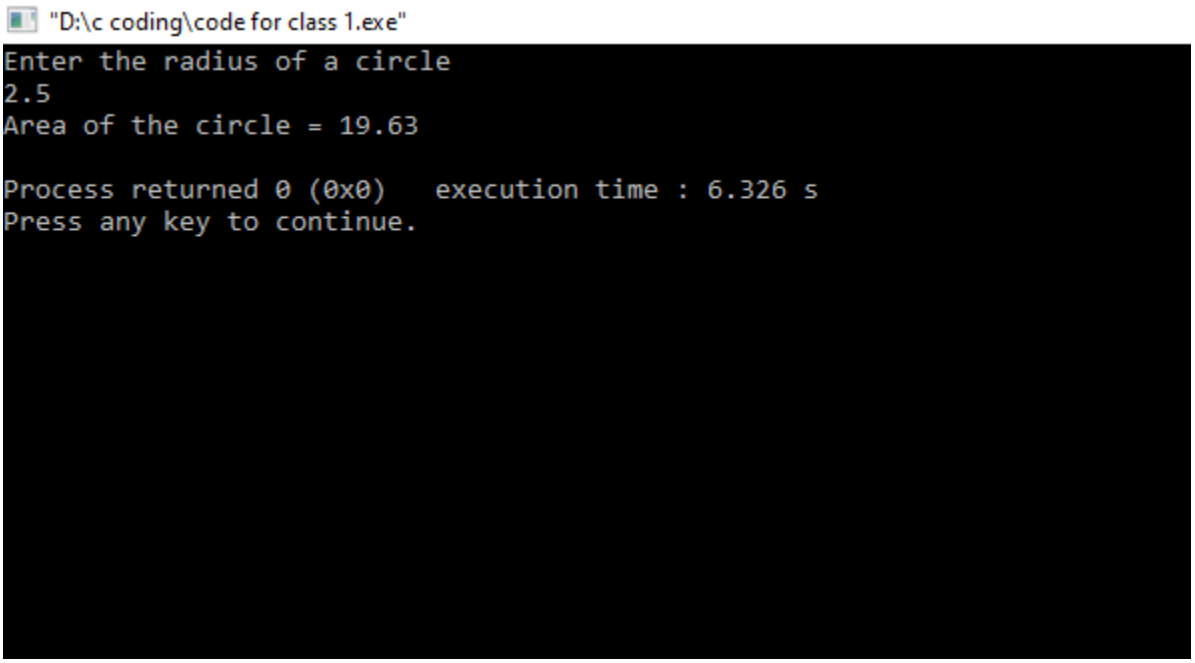
Language used : C

Theory : : We have to print the area of circle via radius entered by the user by using the formula area = 3.14159*radius*radius.

Code:

```
#include <stdio.h>

#include <math.h>

int main()

{

 float radius, area;

 printf("Enter the radius of a circle\n");

 scanf("%f", &radius);

 area = 3.14159*radius*radius;

 printf("Area of the circle = %.2f\n", area);  */ printing upto two decimal place/*

 return 0;

}
```

Output:



(b) Write a C program to calculate simple interest.

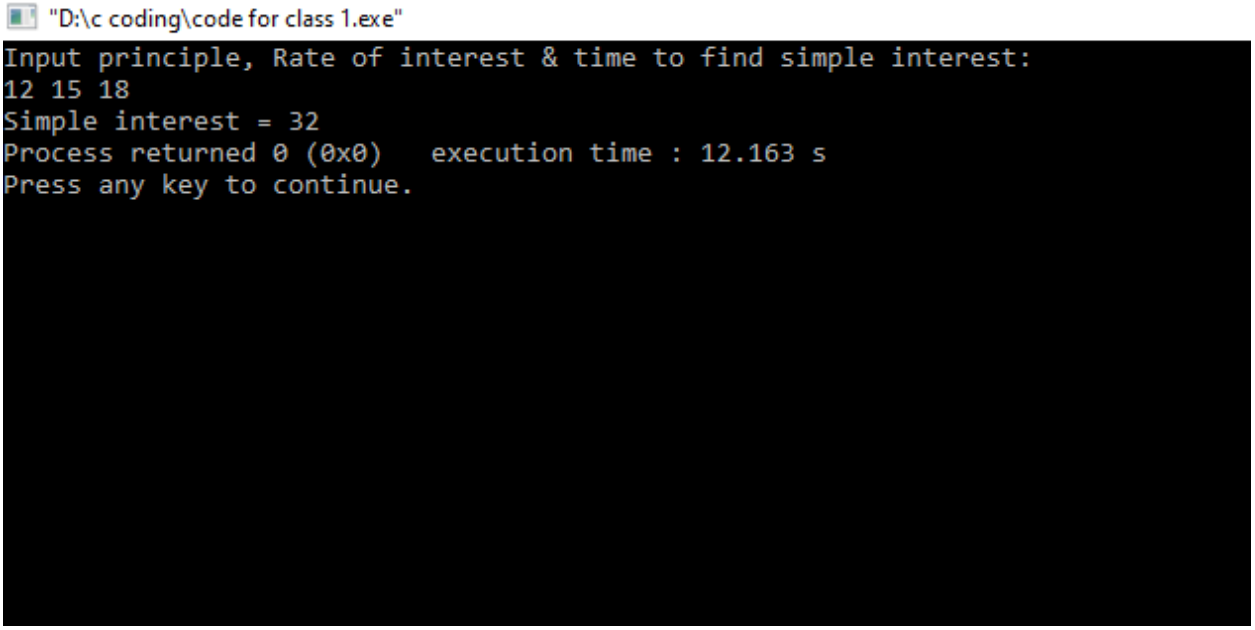
Language used : C

Theory : : We have to print the SI via principle, rate and time entered by the user by using the formula SI=(principle*rate*time)/100.

Code:

```
#include<stdio.h>

int main()

{

   int p,r,t,int_amt;

   printf("Input principle, Rate of interest & time to find simple interest: \n");

   scanf("%d%d%d",&p,&r,&t);

   int_amt=(p*r*t)/100;

   printf("Simple interest = %d",int_amt);

   return 0;

}
```

Output:



3. Write a C program to print a block F using hash (#), where the F has a height of six characters and width of five and four characters.

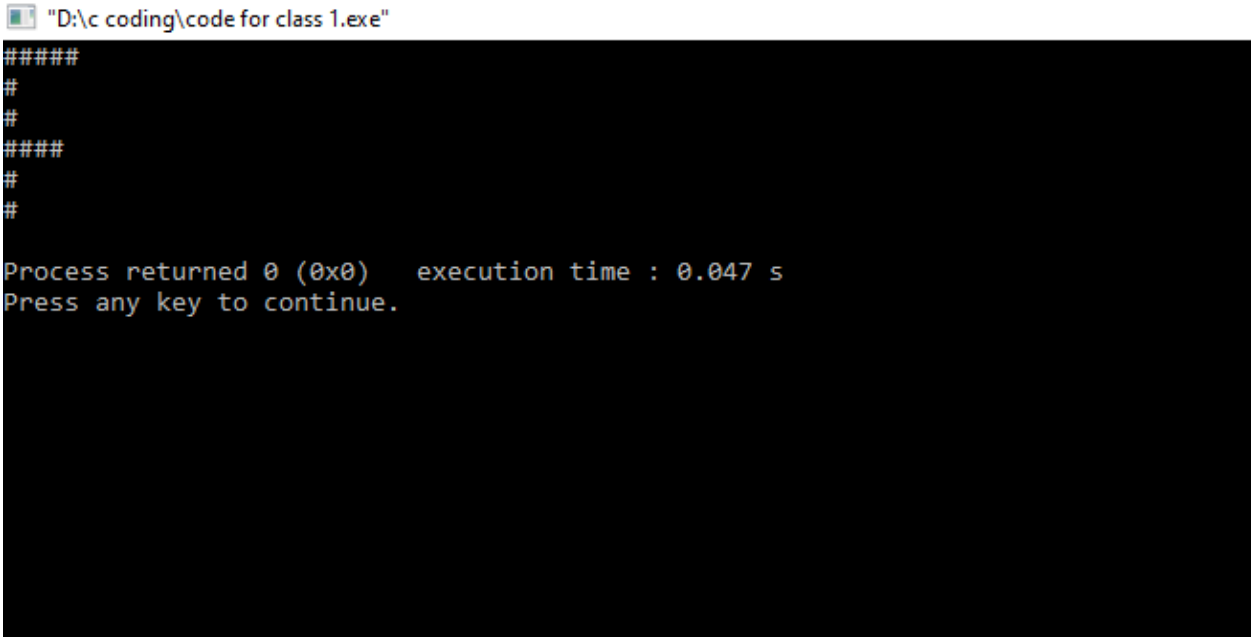
LANGUAGE USED : C

THEORY : WE HAVE TO PRINT THE LETTER 'F' WITH THE HELP OF '#'.

Code:

```
int main()

{

        printf("######\n");

        printf("#\n");

        printf("#\n");

        printf("#####\n");

        printf("#\n");

        printf("#\n");

        printf("#\n");

        return(0);

}
```

Output:



4. Write a C program that accepts two item's weight (floating points' values) and number of purchase (floating points' values) and calculate the average value of the items.

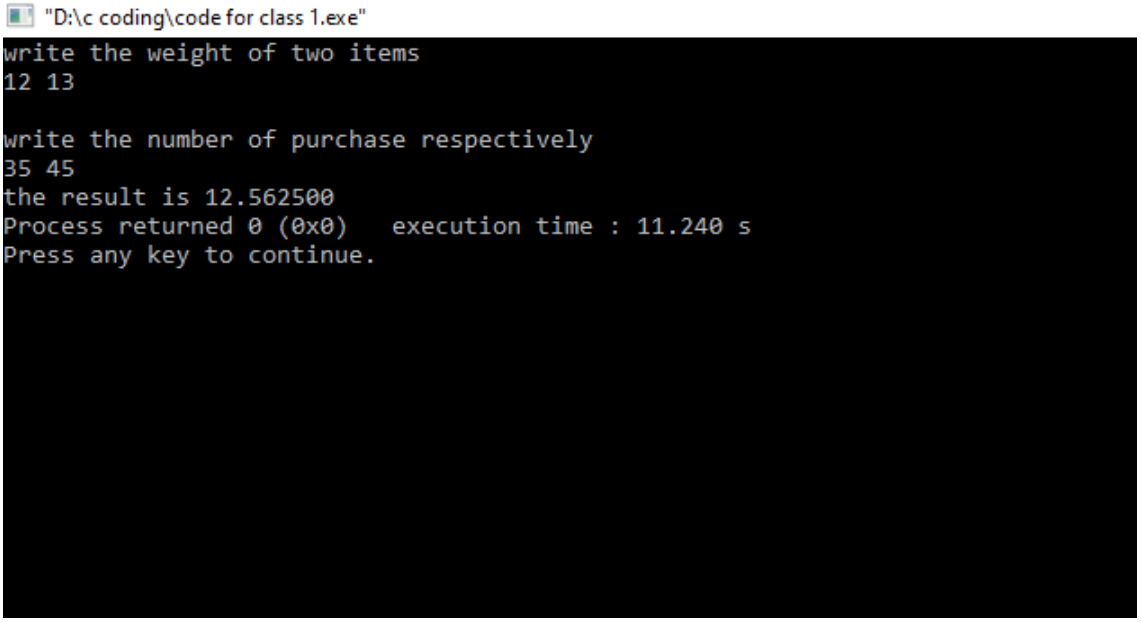
Language used : C

Theory : We have to print the average value of the items via 2 weights variable and number of purchase entered by the user by using formula ((wi1 * ci1) + (wi2 * ci2)) / (ci1 + ci2).

Code:

```
#include<stdio.h>

int main()

{  float wi1, wi2, ci1, ci2, result;

   printf("write the weight of two items \n");

   scanf("%f%f",&wi1, &wi2);

   printf("\n write the number of purchase respectively \n");

   scanf("%f%f",&ci1, &ci2);

   result = ((wi1 * ci1) + (wi2 * ci2)) / (ci1 + ci2);

   printf("the result id %result", result);

   return(0);              }
```

Output:



5. (a) Write a C program to swap two variables using a third variable.

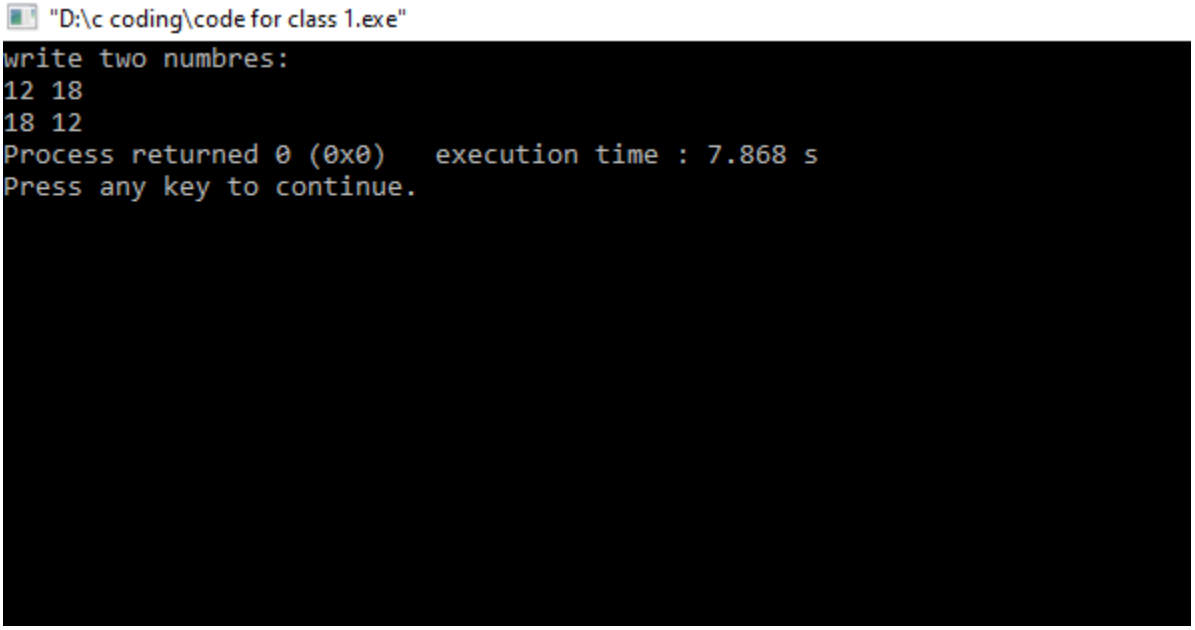
Language used : C

Theory : We have to print the swapped no. via two numbers entered by the user by using c=a, a=b, b=c.

Code:

```
#include<stdio.h>

int main()

{

   int a, b, c;

   printf("write two numbers: \n");

   scanf("%d%d",&a,&b);

   c=a;

   a=b;

   b=c;

   printf("%d %d",a,b);

   return 0;

}
```

Output:



(b) Write a C program to swap two variables without using a third variable.

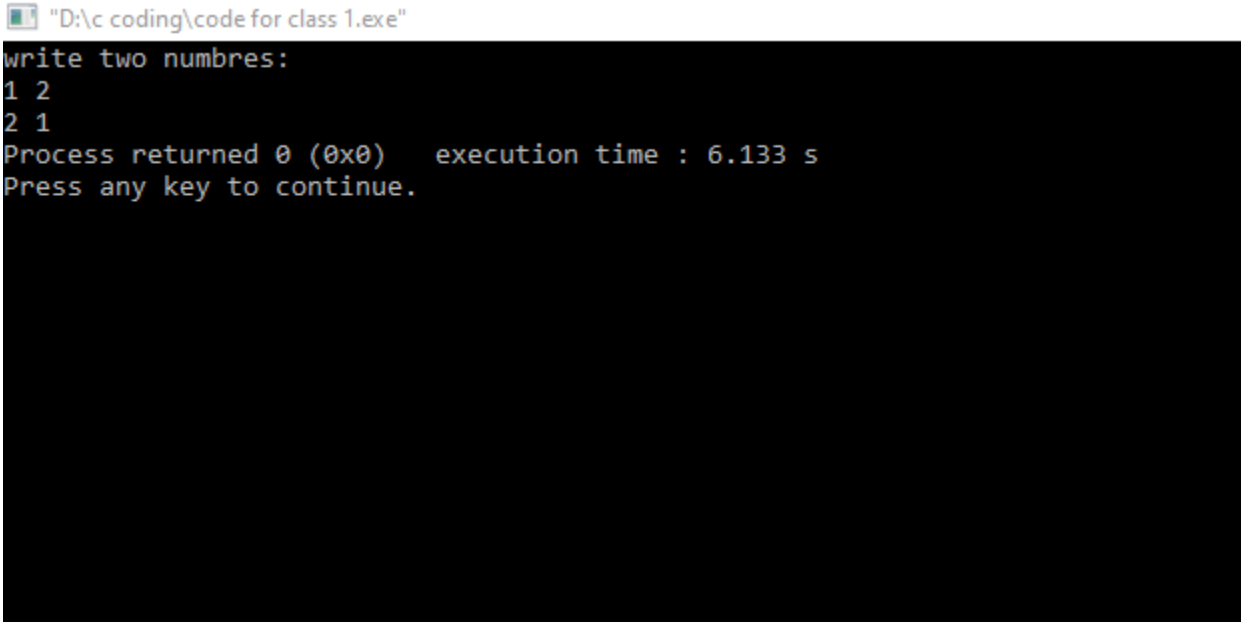
LANGUAGE USED : C

THEORY : WE HAVE TO PRINT THE SWAPPED NO. VIA TWO NUMBERS ENTERED BY THE USER BY USING A=A+B, B=A-B, A=A-.

Code:

```
#include<stdio.h>

int main()

{

   int a, b;

   printf("write two numbres: \n");

   scanf("%d%d",&a,&b);

   a=a+b;

   b=a-b;

   a=a-b;

   printf("%d %d",a,b);

   return 0;

}
```

Output:



6. (a) Write a C program to convert a given integer (in seconds) to hours, minutes, and seconds.

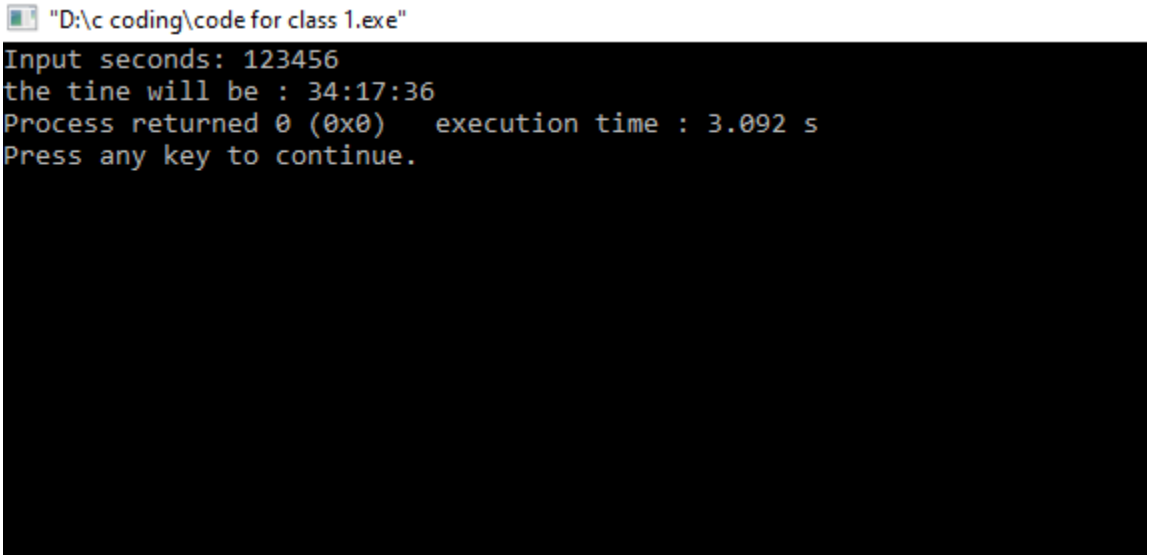
LANGUAGE USED : C

THEORY : WE HAVE TO PRINT HOURS, MINUTES AND SECONDS VIA SECONDS ENTERED BY THE USER BY USING THE FORMULA HOUR=(SECOND/3600), MINUTE=(SECOND-(3600*HOUR))/60, SECOND=(SECOND-(3600*HOUR))-(MINUTE*60).

Code:

```
#include<stdio.h>

int main()

{

   int sec, h, m, s;

   printf("Input seconds: ");

   scanf("%d", &sec);

   h = (sec/3600);

   m = (sec -(3600*h))/60;

   s = (sec -(3600*h)-(m*60));

   printf("the tine will be : %d:%d:%d", h, m, s);

   return 0;

}
```

Output:



(b) Write a C program to convert specified days into years, weeks, and days. Note: Ignore leap year.

Test Data :

Number of days : 1329-3 years,33 weeks and 3 days

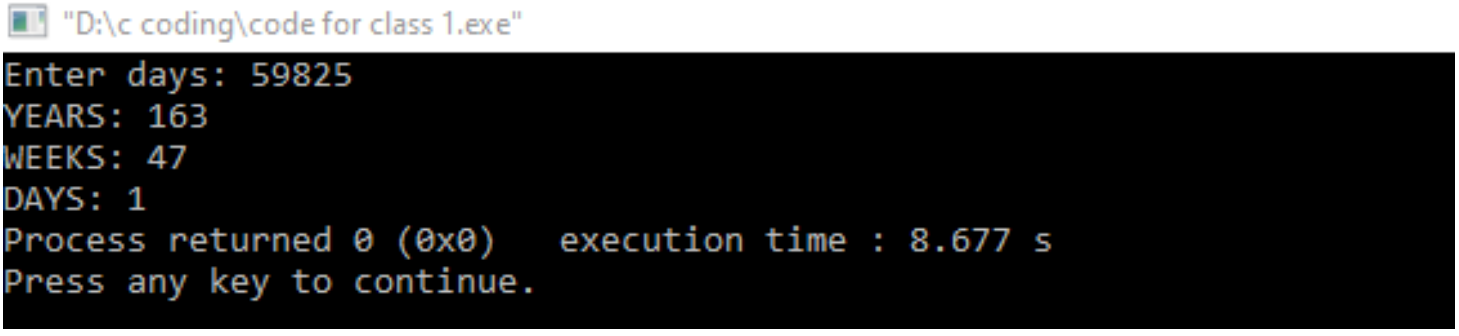
LANGUAGE USED : C

THEORY : WE HAVE TO PRINT INTO YEARS, WEEKS AND DAYS VIA DAYS ENTERED BY THE USER BY USING THE FORMULA YEARS=(DAYS/365), WEEKS=(DAYS%365)/7, DAYS=DAYS-((YEARS*365)+ (WEEKS * 7));

Code:

```
#include <stdio.h>

int main()

{   int days, years, weeks;

    printf("Enter days: ");

    scanf("%d", &days);

    years = (days / 365);   // Ignoring leap year

    weeks = (days % 365) / 7;

    days  = days - ((years * 365) + (weeks * 7));

    printf("YEARS: %d\n", years);

    printf("WEEKS: %d\n", weeks);

    printf("DAYS: %d", days);

    return 0;

}
```

Output:



(c) Write a C program to check whether a number is even or odd.

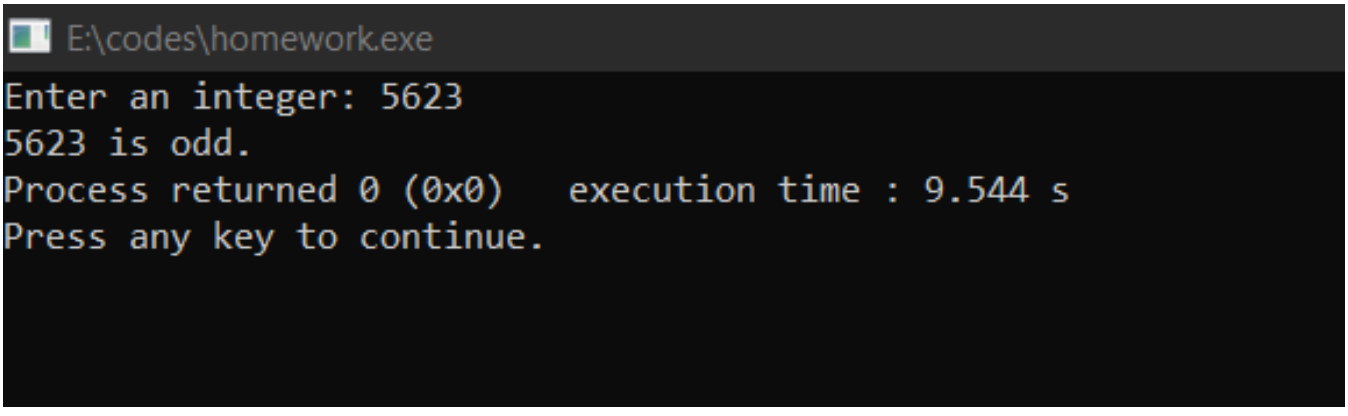
Language used : C

Theory : : We have to print the whether a number is even or odd entered by the user by using the remainder operator and finding the value.

Code:

```
#include <stdio.h>

int main() {

   int num;

   printf("Enter an integer: ");

   scanf("%d", &num);

   // True if num is perfectly divisible by 2

   if(num % 2 == 0)

      printf("%d is even.", num);

   else

      printf("%d is odd.", num);

   return 0;

}
```

Output:



7. Write a C program to check whether a given year is Leap year or not.

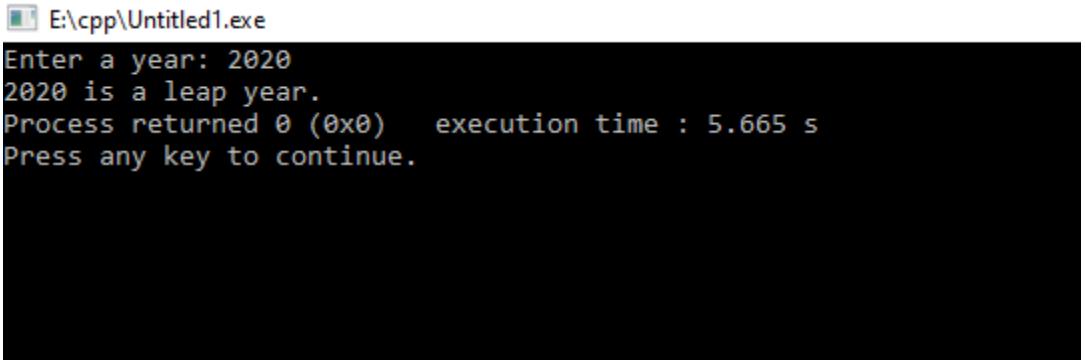
Language used : C

Theory : We have to print whether the year is a leap year or not via year entered by the user by using the remainder operator and if else condition.

Code:

```
#include <stdio.h>

int main()

{

  int year;

  printf("Enter a year: ");

  scanf("%d", &year);

  if (year % 400 == 0)

    printf("%d is a leap year.", year);

  else if (year % 100 == 0)

    printf("%d is not a leap year.", year);

  else if (year % 4 == 0)

    printf("%d is a leap year.", year);

  else

    printf("%d is not a leap year.", year);

  return 0;

}
```

Output:



8. (a) Write a C program to check whether a triangle is Equilateral, scalene, or isosceles.

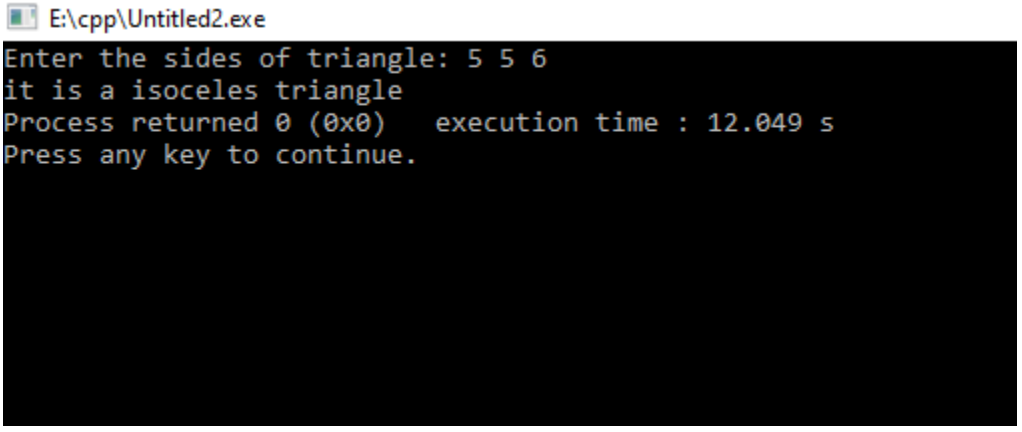
Language used : C

Theory : We have to print whether the triangle is Equilateral, scalene or isosceles or not via three sides entered by the user by using the && and || operands and comparing values.

Code:

```
#include <stdio.h>

int main()

{

  int a, b, c;

  printf("Enter the sides of triangle: ");

  scanf("%d%d%d", &a, &b, &c);

  if (a==b&&a==c)

    printf("it is a equiletral triangle");

  else if (a==b||a==c||b==c)

    printf("it is a isoceles triangle");

  else

    printf("it is a scelene triangle");

  return 0;

}
```

Output:



(b) Write a C program to check whether a triangle is right angles, obtuse, acute triangle Equilateral, scalene, or isosceles.

Language used : C

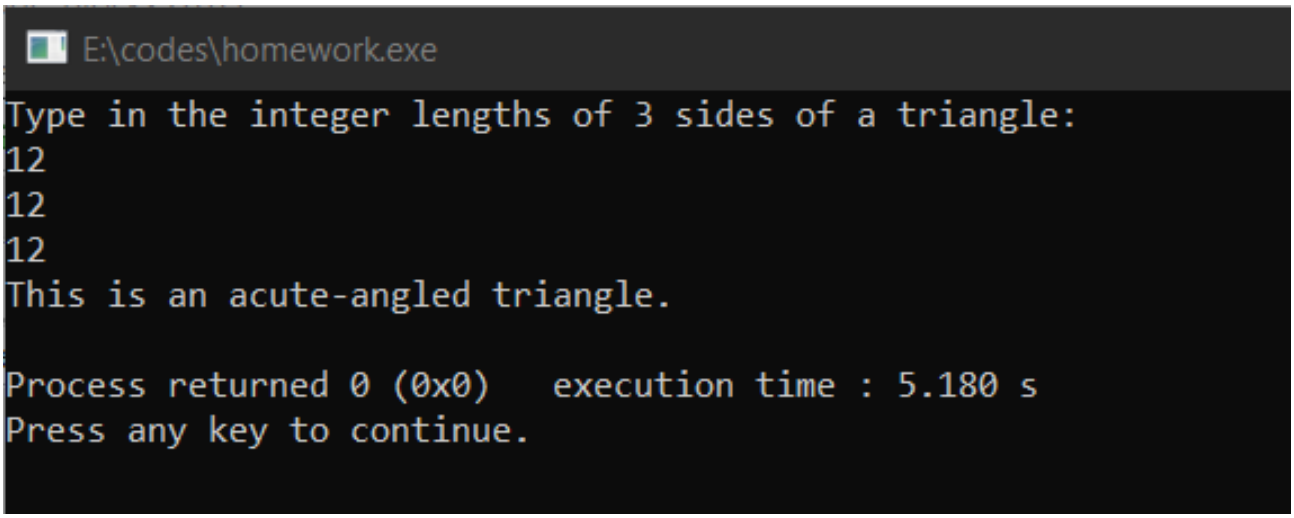
Theory : We have to print whether the triangle is Equilateral, scalene or isosceles or not via three sides entered by the user by using OR(||) operator and if else condition.

Code:

```
#include <stdio.h>

#include <stdlib.h>

#include <math.h>

int main(int argc, char *argv[])

{  int x,y,z;

   printf("Type in the integer lengths of 3 sides of a triangle:\n");

   scanf("%d %d %d", &x, &y, &z);

   if((x<=0) || (y<=0) || (z<=0))

        printf("This is not a triangle.\n");

   else

     {

     if((x + y <= z) || (x + z <= y) || (y + z <= x))

        printf("This is not a triangle.\n");

     else

        if( ((x * x) + (y * y) == (z * z)) || ((x * x) + (z * z) == (y * y)) || ((z * z) + (y * y) == (x * x)) )

           printf("This is a right-angled triangle.\n");

        else if( ( ((x * x) + (y * y) < (z * z)) || ((x * x) + (z * z) < (y * y)) || ((z * z) + (y * y) < (x * x)) ) || ( ( x<=z
&& y<=z ) || ( x<=y && z<=y ) || ( y<=x && z<=x ) ) )

           printf("This is an acute-angled triangle.\n");

        else if( ( ((x * x) + (y * y) > (z * z)) || ((x * x) + (z * z) > (y * y)) || ((z * z) + (y * y) > (x * x)) ) || ( ( x>z &&
y>z ) || ( x>y && z>y ) || ( y>x && z>x ) ) )
```

```
            printf("This is an obtuse-angled triangle.\n");
        else
            printf("Not a triangle\n");
    }
  return 0;
}
```

Output:



9. Write a C program to covert temperature from Fahrenheit to Celsius and Celsius to Fahrenheit (User must provide the choice of type of temperature).

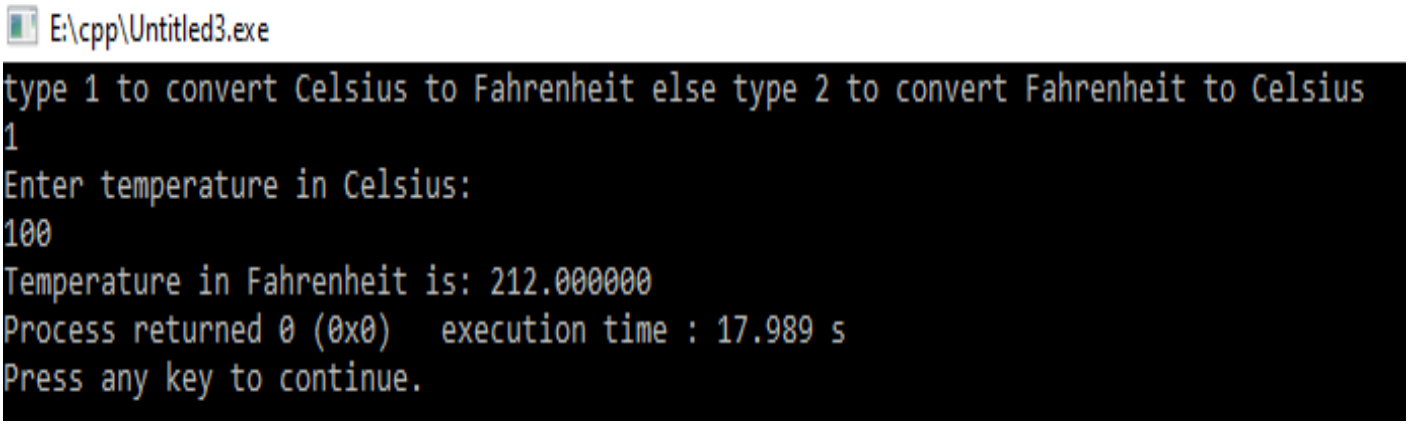
LANGUAGE USED : C

THEORY : WE HAVE TO PRINT TEMPERATURE FROM FAHRENHEIT TO CELSIUS AND CELSIUS TO FAHRENHEIT ENTERED BY THE USER BY USING THE FORMULA F = (1.8*C) + 32 AND C= (F-32) / 1.8.

Code:

```
#include<stdio.h>

int main()

{  int a;

   float c, f;

   printf("type 1 to convert Celsius to Fahrenheit else type 2 to convert Fahrenheit to Celsius \n");

   scanf("%d",&a);

   if(a==1)

   {  printf("Enter temperature in Celsius: \n");

      scanf("%f", &c);

      f = (1.8*c) + 32;

      printf("Temperature in Fahrenheit is: %f ", f);

   }

   else if(a==2)

   {

       printf("Enter temperature in Fahrenheit: \n");

       scanf("%f", &f);

       c= (f-32) / 1.8;

       printf("Temperature in Celsius is: %f ", c); }

   return 0;}.
```

Output:


E:\cpp\Untitled3.exe
type 1 to convert Celsius to Fahrenheit else type 2 to convert Fahrenheit to Celsius
1
Enter temperature in Celsius:
100
Temperature in Fahrenheit is: 212.000000
Process returned 0 (0x0) execution time : 17.989 s
Press any key to continue.

10. (a) Write a C program to check whether a character is an alphabet, digit.

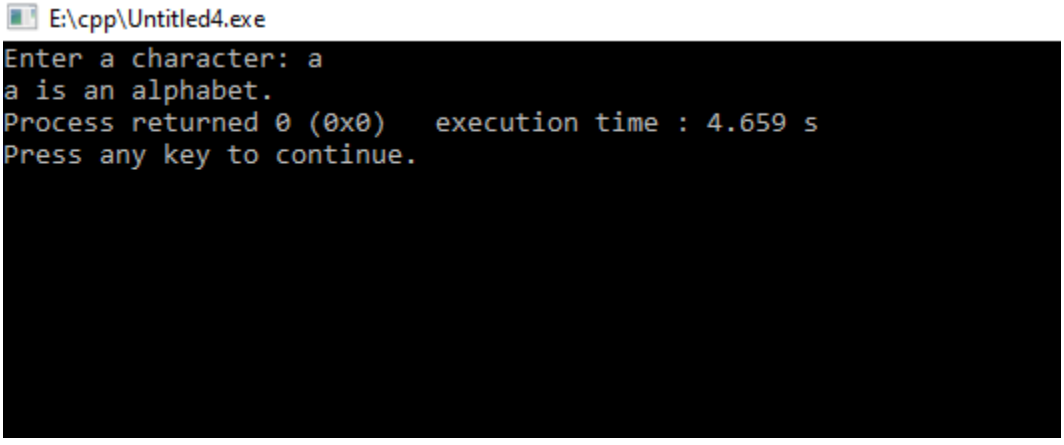
Language used : C

Theory : : We have to print whether the character is an alphabet or digit via variable entered by the user by using the comparison operand OR(||) and if else condition.

Code:

```
#include <stdio.h>

int main()

{

   char c;

   printf("Enter a character: ");

   scanf("%c", &c);

   if ((c >= 'a' && c <= 'z') || (c >= 'A' && c <= 'Z'))

      printf("%c is an alphabet.", c);

   else if(c>='1'&& c<='9')

      printf("%c is a number.", c);


   return 0;

}
```

Output:



(b) Write a C program a program to check whether an alphabet is a vowel or consonant

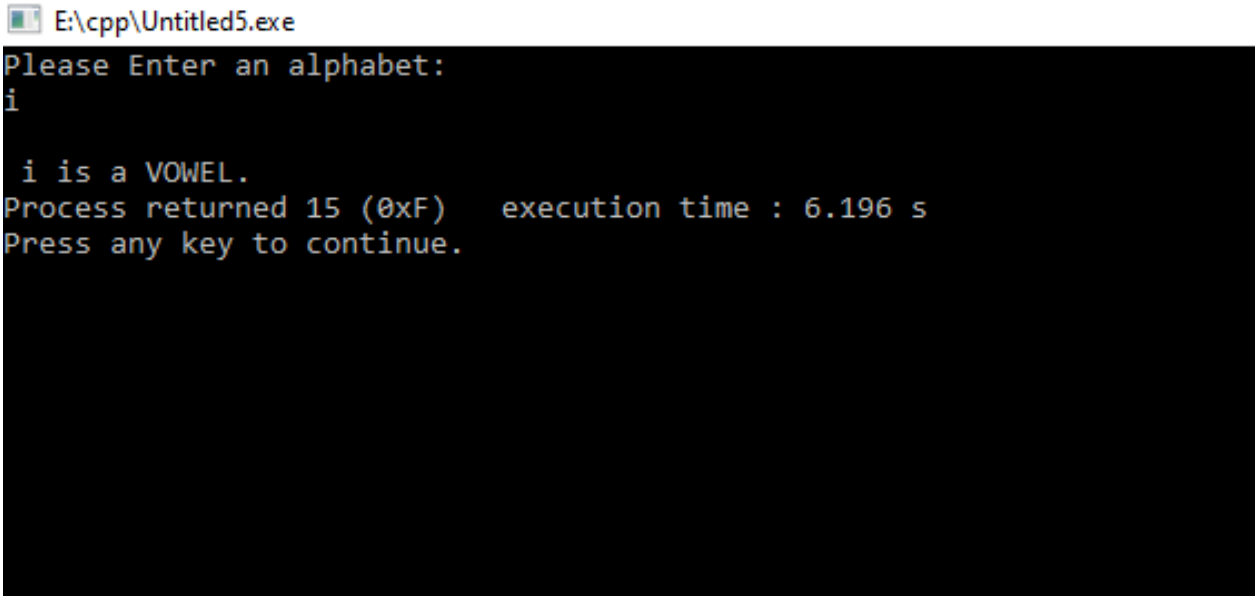
Language used : C

Theory : We have to print whether the character is an alphabet is a vowel or consonant via variable entered by the user by using the comparison operand OR(||) and if else condition.

Code:

```
#include <stdio.h>

void main()

{

   char c;

   printf("Please Enter an alphabet: \n");

   scanf(" %c", &c);

   if ((c == 'a' || c == 'e' || c == 'i' || c == 'o' || c == 'u') || (c == 'A' || c == 'E' || c == 'I' || c == 'O' || c == 'U'))

         printf("\n %c is a VOWEL.", c);

   else

         printf("\n %c is a CONSONANT.", c);

}
```

Output:



11. (a) Write a C program to find smallest of two numbers.

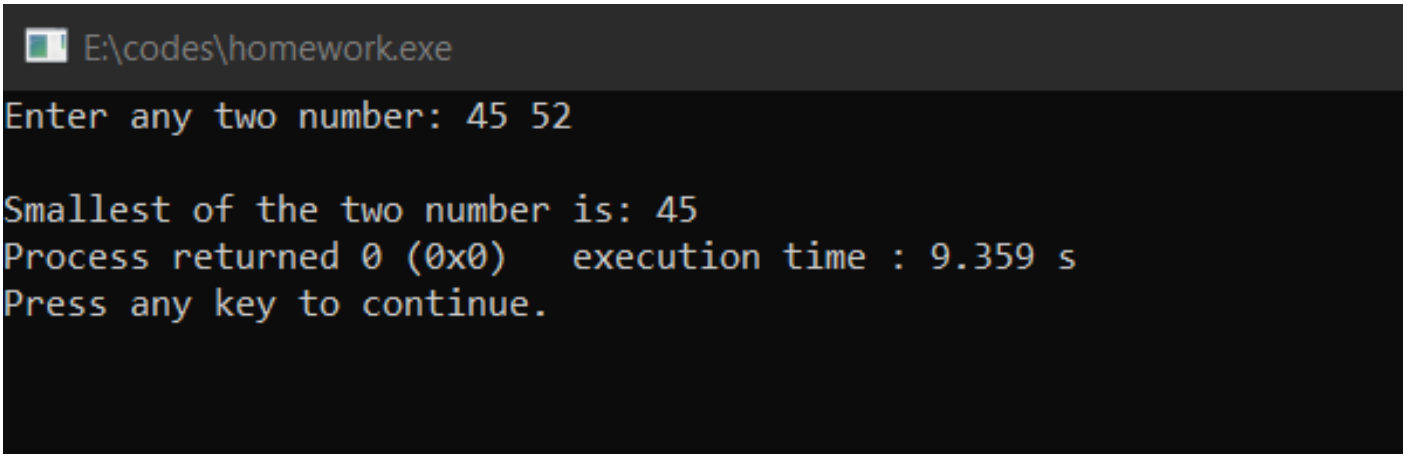
Language used : C

Theory : : We have to print smallest of two numbers via two variables entered by the user by using if else condition.

Code:

```
#include<stdio.h>

int main()

{

   int a, b, small;

   printf("Enter any two number: ");

   scanf("%d%d", &a, &b);

   if(a<b)

      small=a;

   else

      small=b;

   printf("\nSmallest of the two number is: %d", small);

   return 0;

}
```

Output:



(b) Write a C program to find largest of three numbers

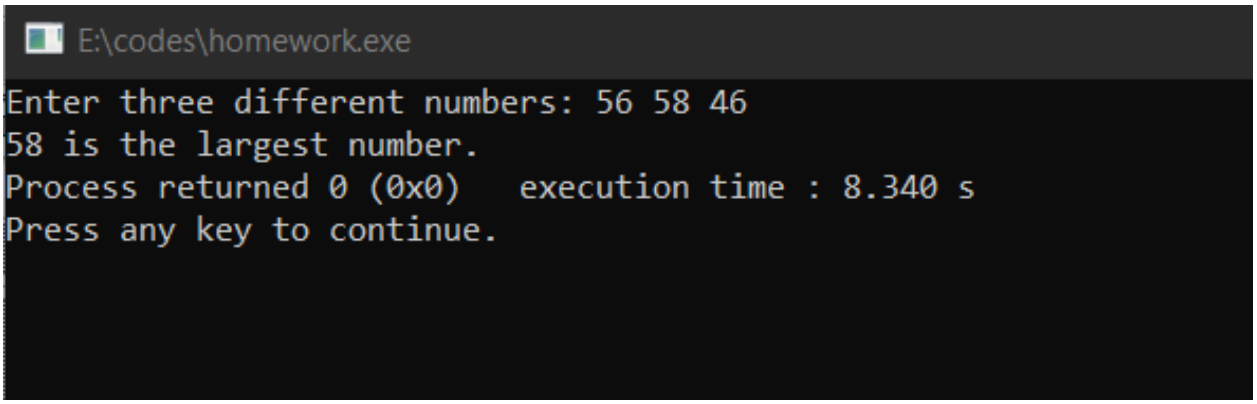
LANGUAGE USED : C

THEORY: WE HAVE TO PRINT THE LARGEST OF THREE NUMBERS VIA THREE NUMBERS ENTERED BY THE USER BY USING IF CONDITION AND COMPARING OPERATORS.

Code:

```
#include <stdio.h>

int main() {

   int n1, n2, n3;

   printf("Enter three different numbers: ");

   scanf("%d %d %d", &n1, &n2, &n3);

   if (n1 >= n2 && n1 >= n3)

      printf("%d is the largest number.", n1);

   if (n2 >= n1 && n2 >= n3)

      printf("%d is the largest number.", n2);

   if (n3 >= n1 && n3 >= n2)

      printf("%d is the largest number.", n3);


   return 0;

}
```

Output:



12. Write a program in C to implement Simple Calculator.

Language used : C

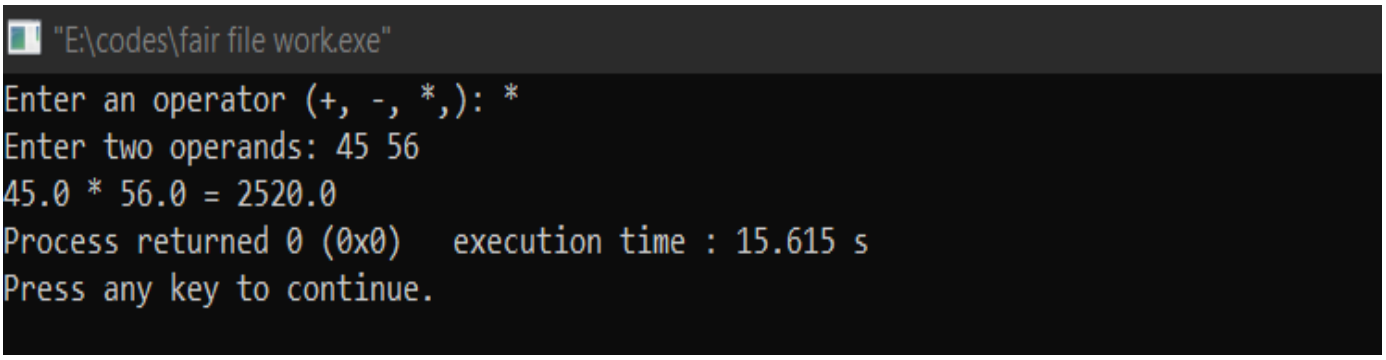
Theory: We have to print the operation of simple calculator by two numbers entered by the user by using the switch and simply do addition, subtraction, multiplication and division.

Code:

```
#include <stdio.h>

int main()

{   char operator;

   double first, second;

   printf("Enter an operator (+, -, *,): ");

   scanf("%c", &operator);

   printf("Enter two operands: ");

   scanf("%lf %lf", &first, &second);

   switch (operator)

   {  case '+':

         printf("%.1lf + %.1lf = %.1lf", first, second, first + second);

            break;

      case '-':

         printf("%.1lf - %.1lf = %.1lf", first, second, first - second);

            break;

      case '*':

         printf("%.1lf * %.1lf = %.1lf", first, second, first * second);

            break;

      case '/':

         printf("%.1lf / %.1lf = %.1lf", first, second, first / second);

            break;

      default:

         printf("Error! operator is not correct");
```

```
   }
return 0;
}
```

Output:



13. WAP to calculate the root of a Quadratic Equation.

Language used : C

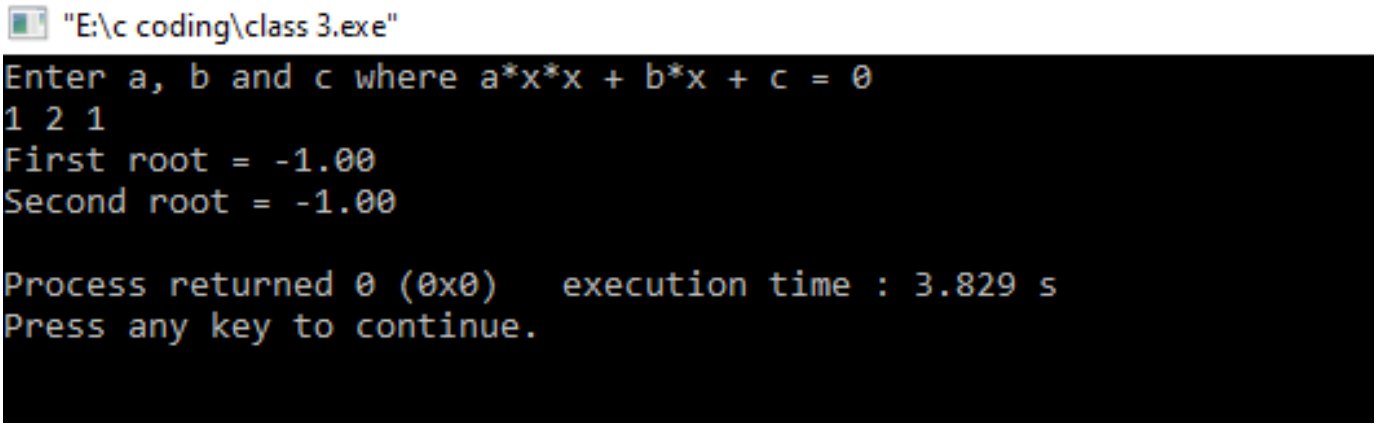
Theory : We have to print the root of Quad. Eqn. via three no. entered by the user by using d = b*b - 4*a*c and root1 = (-b + sqrt(d))/(2*a), root2 = (-b - sqrt(d))/(2*a).

Code:

```
#include <stdio.h>

#include <math.h>

int main()

{

 int a, b, c, d;

 double root1, root2;


 printf("Enter a, b and c where a*x*x + b*x + c = 0\n");

 scanf("%d%d%d", &a, &b, &c);


 d = b*b - 4*a*c;


 if (d < 0) { //complex roots

  printf("First root = %.2lf + i%.2lf\n", -b/(double)(2*a), sqrt(-d)/(2*a));

  printf("Second root = %.2lf - i%.2lf\n", -b/(double)(2*a), sqrt(-d)/(2*a));

 }

 else { //real roots

  root1 = (-b + sqrt(d))/(2*a);

  root2 = (-b - sqrt(d))/(2*a);


  printf("First root = %.2lf\n", root1);
```

```
    printf("Second root = %.2lf\n", root2);

  }


  return 0;

}
```

Output:



14. WAP to to accept a coordinate point in a XY coordinate system and determine in which quadrant the coordinate point lies.

LANGUAGE USED : C

THEORY: WE HAVE TO PRINT IN WHICH QUADRANT THE COORDINATE POINT LIES VIA COORDINATES ENTERED BY THE USER BY USING IF ELSE CONDITIONS.

Code:

```
#include <stdio.h>

void main() {

        int co1,co2;

        printf("Input the values for X and Y coordinate : ");

        scanf("%d %d",&co1,&co2);

        if( co1 > 0 && co2 > 0)

          printf("The coordinate point (%d,%d) lies in the First quandrant.\n",co1,co2);

        else if( co1 < 0 && co2 > 0)

          printf("The coordinate point (%d,%d) lies in the Second quandrant.\n",co1,co2);

        else if( co1 < 0 && co2 < 0)

          printf("The coordinate point (%d, %d) lies in the Third quandrant.\n",co1,co2);

        else if( co1 > 0 && co2 < 0)

          printf("The coordinate point (%d,%d) lies in the Fourth quandrant.\n",co1,co2);

        else if( co1 == 0 && co2 == 0)

          printf("The coordinate point (%d,%d) lies at the origin.\n",co1,co2);

}
```

Output:

15. Write a program to find gross salary of employee if DA is 40% of basic Salary and HRA is 20% of basic salary. Basic salary will be entered as input by keyboard.

Language used : C

Theory: We have to find gross salary of employee if DA is 40% of basic Salary and HRA is 20% of basic salary via Basic salary entered by the user by using the formula gross_salary=1.6*basic_salary.

Code:

```
#include <stdio.h>

int main()

{

  float da, hra, basic_salary, gross_salary;



  printf("Enter your basic salary: ");

  scanf("%f", &basic_salary);



  da = 0.4*basic_salary;

  hra=0.2*basic_salary;

  gross_salary = basic_salary+da+hra;



  printf("Your gross salary is %f.", gross_salary);

  return 0;

}
```

Output:

```
E:\codes\homework.exe
Enter your basic salary: 25000
Your gross salary is 40000.000000.
Process returned 0 (0x0)   execution time : 116.681 s
Press any key to continue.
```

16. Write a program in C to calculate and print the Electricity bill of a given customer. The customer id and unit consumed by the user should be taken from the keyboard and display the total amount to pay to the customer.
upto 199----------1.20 200-500----------1.80
Above 500----------2.00 If bill exceeds Rs. 400
then a surcharge of 15% will be charged and the minimum bill should be of Rs. 100/-

Language used : C

Theory: We have to print THE ELECTRICITY BILL via below condition

upto 199----------1.20
200-500----------1.80
Above 500----------2.00
If bill exceeds Rs. 400 then a surcharge of 15% will be charged and the minimum bill should be of Rs. 100/-

Code:

```
#include <stdio.h>

#include <string.h>

void main()

{  int custid, conu;

   float chg, surchg=0, gramt,netamt;

   char connm[25];

   printf("Input Customer ID :");

   scanf("%d",&custid);

   printf("Input the name of the customer :");

   scanf("%s",connm);

   printf("Input the unit consumed by the customer : ");

   scanf("%d",&conu);

   if (conu <200 )

         chg = 1.20;

   else   if (conu>=200 && conu<=500)
```

```
                chg = 1.50;
        else if (conu>=400 && conu<600)
                        chg = 1.80;
                else
                        chg = 2.00;
  gramt = conu*chg;
  if (gramt>400)
        surchg = gramt*15/100.0;
  netamt = gramt+surchg;
  if (netamt  < 100)
        netamt =100;
  printf("\nElectricity Bill\n");
  printf("Customer IDNO                :%d\n",custid);
  printf("Customer Name                :%s\n",connm);
  printf("unit Consumed                :%d\n",conu);
  printf("Amount Charges @Rs. %4.2f  per unit :%8.2f\n",chg,gramt);
  printf("Surchage Amount              :%8.2f\n",surchg);
  printf("Net Amount Paid By the Customer    :%8.2f\n",netamt);
}
```

Output:

"E:\c coding\class 3.exe"

```
Input Customer ID :52
Input the name of the customer :harsh
Input the unit consumed by the customer : 600

Electricity Bill
Customer IDNO                       :52
Customer Name                       :harsh
unit Consumed                       :600
Amount Charges @Rs. 2.00  per unit : 1200.00
Surchage Amount                     :  180.00
Net Amount Paid By the Customer     : 1380.00

Process returned 46 (0x2E)   execution time : 14.645 s
Press any key to continue.
```

17. A library charges a fine for every book returned late. For first 5 days the fine is 50 paisa, for 6-10 days, fine is one rupee and above 10 days, fine is 5 rupees. If you return the book after 30 days your membership will be cancelled. Write a program to accept the number of days the member is late to return the book and display the fine or appropriate message.

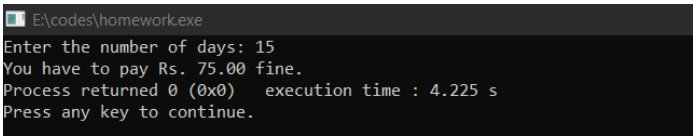
LANGUAGE USED : C

THEORY: WE HAVE TO PRINT THE FINE OR APPROPRIATE MESSAGE BY FINE ENTERED BY THE USER AND BY USING THE CORRECT MULTIPLE IF CONDITION.

Code:

```
#include<stdio.h>

#include<conio.h>

int main()

{  int days;

   float fine;

   printf("Enter the number of days: ");

   scanf("%d", &days);

   if (days > 0 && days <= 5)

      fine = 0.50 * days;

   if (days >= 6 && days <= 10)

      fine = 1 * days;

   if (days > 10)

      fine = 5 * days;

   if (days > 30)

        printf("Your membership would be canceled.\n");

   printf("You have to pay Rs. %.2f fine.", fine);

return0;}
```

Output:



18. Write a program to find the factorial of any number.

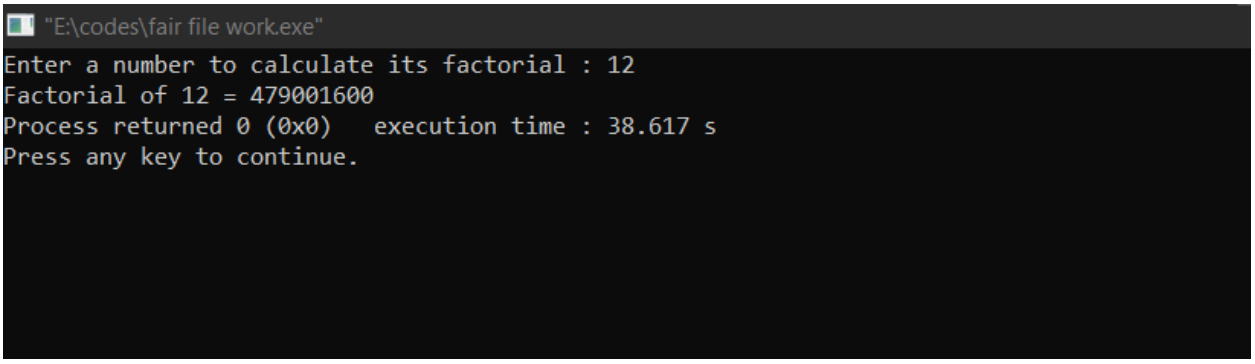
Language used : C

Theory: We have to print the factorial by one number entered by the user by using for loop.

Code:

```
#include <stdio.h>

int main() {

   int i , num , fact = 1;

   printf ("Enter a number to calculate its factorial : ");

   scanf ("%d", &num);

   if (num<0). {

        printf ("Factorial is not defined for negative numbers.");

     }

   else  {

        for(i=1;i<= num;i++) {

             fact = fact * i;

          }

        printf("Factorial of %d = %d", num, fact);

     }

   return 0;

}
```

Output:



19. Write a program to print Fibonacci sequence 0 1 1 2 3 5 8 13..... N terms and prints the sum of sequence.

LANGUAGE USED : C

THEORY: WE HAVE TO PRINT THE FIBONACCI SEQUENCE 0, 1, 1, 2, 3, 5, 8, 13.....N TERMS AND PRINTS THE SUM OF SEQUENCE BY ONE NUMBER ENTERED BY THE USER BY USING WHILE.

Code:

```
#include <stdio.h>
#include <math.h>
int main()
{    int f1,f2,f3,n,i=2,s=1;
     f1=0;
     f2=1;
     printf("How many terms do you \nwant in Fibonacci series? : ");
     scanf("%d",&n);
     printf("\nFibonacci Series Upto %d Terms:\n\n",n);
     printf("%d, %d",f1,f2);
     while(i<n)
     {
          f3=f1+f2;
          printf(", %d",f3);
          f1=f2;
          f2=f3;
          s=s+f3;
          i++;
     }
     printf("\n\nSum of Fibonacci Series : %d",s);
     return 0;}
```

Output:

```
"E:\codes\fair file work.exe"
number of terms you want in Fibonacci series? : 15

Fibonacci Series Upto 15 Terms:

0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377

Sum of Fibonacci Series : 986
Process returned 0 (0x0)   execution time : 2.536 s
Press any key to continue.
```

20. Write a program in C to accept an integer numbers and find sum of digits.

Language used: C

Theory: user is required to enter the number. When we divide a number by 10, it will leave a remainder as its last digit, then after taking the remainder from the quotient above is the second digit and so on; in this way we take the concept of loops to calculate the sum of digits of a number.

CODE:

```
#include <stdio.h>

int main() {

  int n, t, sum = 0, remainder;

  printf("Enter an integer\n");

  scanf("%d", &n);

  t = n;

  while (t != 0)

  {

    remainder = t % 10;

    sum = sum + remainder;

    t = t / 10;

  }

  printf("Sum of digits of %d = %d\n", n, sum);

  return 0;

}
```

OUTPUT:

```
Enter an integer
202
Sum of digits of 202 = 4
```

21. Write a program in C to accept an integer numbers and find reverse of this number and check this number for palindrome.

LANGUAGE USED: C

THEORY: WE HAVE TO PRINT THE REVERSE OF THIS NUMBER AND CHECK IF THIS NUMBER IS PALINDROME BY ONE NUMBER ENTERED BY THE USER BY USING WHILE CONDITION.

CODE:

```
#include <stdio.h>

void main() {

   int num, temp, remainder, reverse = 0;

   printf("Enter an integer number \n");

   scanf("%d", &num);

   /*  original number is stored at temp */

   temp = num;

   while (num > 0)  {

      remainder = num % 10;

      reverse = reverse * 10 + remainder;

      num /= 10;

   }

   printf("Given number is = %d\n", temp);

   printf("Its reverse is  = %d\n", reverse);

   if (temp == reverse)

      printf("Number is a palindrome \n");

   else

      printf("Number is not a palindrome \n");}
```

OUTPUT:

```
Enter an integer number
1001
Given number is = 1001
Its reverse is  = 1001
Number is a palindrome
```

22. Write a program in C to accept an integer numbers and to check a number is Armstrong or not

Language used: C

Theory: Armstrong is the number in which the sum of cubes of the digits of a number is equal to the number itself. So, We have to check a number is Armstrong or not by one number entered by the user by using while condition and if else.

CODE:

```
#include <stdio.h>

int main()

{

   int num, originalNum, remainder, result = 0;

   printf("Enter a three-digit number: ");

   scanf("%d", &num);

   originalNum = num;

   while (originalNum != 0)

   {

     // remainder contains the last digit

      remainder = originalNum % 10;

     result += remainder * remainder * remainder;

     // removing last digit from the orignal number

     originalNum /= 10;

   }

   if (result == num)

      printf("%d is an Armstrong number.", num);

   else

      printf("%d is not an Armstrong number.", num);

   return 0;

}
```

OUTPUT:

```
Enter a three-digit number: 153
153 is an Armstrong number.
```

```
Enter a three-digit number: 152
152 is not an Armstrong number.
```

23. Write a program in C to accept an integer numbers and to check a number is Perfect or not

LANGUAGE USED: C

THEORY: PERFECT NUMBER IS THE NUMBER IN WHICH THE SUM OF ITS POSITIVE DIVISORS IS EQUAL TO ITSELF. SO, WE HAVE TO CHECK A NUMBER IS PERFECT OR NOT BY ONE NUMBER ENTERED BY THE USER BY USING FOR LOOP AND IF ELSE.

CODE:

```
#include <stdio.h>

int main()

{

   int i, num, sum = 0;

   /* Input a number from user */

   printf("Enter any number to check perfect number: ");

   scanf("%d", &num);

   /* Calculate sum of all proper divisors */

   for(i = 1; i <= num / 2; i++)

   {

      /* If i is a divisor of num */

      if(num%i == 0)

      {

         sum += i;

      }

   }

  /* Check whether the sum of proper divisors is equal to num */
```

```
    if(sum == num)
    {
        printf("%d is PERFECT NUMBER", num);
    }
    else
    {
        printf("%d is NOT PERFECT NUMBER", num);
    }
    return 0;
}
```

OUTPUT:

```
Enter any number to check perfect number: 6
6 is PERFECT NUMBER
```

24. Write a program to find the sum of following series: S = 2+4+6+8+.....N terms

LANGUAGE USED- C

THEORY- WE HAVE TO PRINT THE SUM OF SERIES UPTO N TERMS ENTERED BY THE USER BY USING FOR LOOP AND FLAG.

CODE:

```
#include<stdio.h>

int main()

{

 int sum=0,k=2,n,i;

 printf("Enter the value of n ");

 scanf("%d",&n);

 printf("Series: ");

 for(i=1; i<=n; i++)

  {

   printf("%d ",k);

   sum= sum+k;

   k=k+2;

  }

   printf("\nSum: %d ",sum);

   return 0;

}
```

OUTPUT:

```
Enter the value of n 7
Series: 2 4 6 8 10 12 14
Sum: 56
```

25. Write a program to check a number whether it is prime number or not.

Language used: C

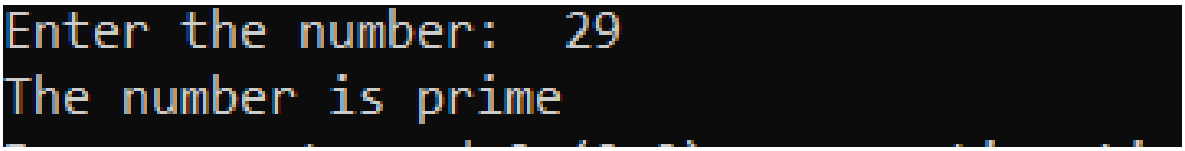
Theory: First we will check if the number is 1 or not, if it is 1 then it is neither prime nor composite; else we will check if it is divisible by terms other than 1 and itself then it is prime otherwise not.

CODE:

```
#include<stdio.h>

int main()

{

  int n,i,a,c;

  printf("Enter the number:  ");

  scanf("%d",&n);

  if (n==1)

  {

   printf("The number is neither prime nor composite");

  }

   else

   {

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

    {

     a= n%i;

     if (a==0)

     {

       c=0;

       break;

     }

     else
```

```
    {
     c=2;
    }
   }
  if(c==0)
  {
  printf("\nThe number is composite");
  }
  else
  {
   printf("The number is prime");
  }

  }
   return 0;
}
```

OUTPUT:



26. Write a program to find the sum of following series: 1 – 1/2 + 1/3 – 1/4 + 1/5 - ..... up to n terms.

LANGUAGE USED: C

THEORY: USER IS SUPPOSED TO ENTER THE VALUE OF N, WE WILL USE THE CONCEPT OF FOR LOOP TO PRINT THIS SERIES.

CODE:

```
#include<stdio.h>

#include<math.h>

int main(void)

{

   int n,i;

   float sum=0;

   printf("Enter the number of terms ");

   scanf("%d",&n);

   for(i=1; i<=n; ++i) {

      sum= sum+ (pow(-1,i+1))/i;}

   printf("%f",sum);

}
```

OUTPUT:

```
Enter the number of terms 4
0.583333
```

27. Write a program to find the sum of following series: 1! + 2! + 3! + 4! + ... + n!

LANGUAGE USED: C

THEORY: USER IS SUPPOSED TO ENTER THE VALUE OF N, WE WILL USE THE CONCEPT OF LOOPS TO PRINT THIS SERIES.

CODE:

```
#include <stdio.h>

void main()

{

   int n,f,i,j,sum=0;

   printf("Enter the Number of terms: ");

   scanf("%d",&n);

   for(i=1;i<=n;i++)

   {

      f = 1;

      for(j=i; j>0; j--)

      {

        f = f*j;

      }

      sum = sum + f;

   }

   printf("The sum of factorials upto %d terms is %d", n,sum);

}
```

OUTPUT:

```
Enter the Number of terms: 5
The sum of factorials upto 5 terms is 153
```

28. Write a program to find the sum of following series: S = $-1^3+3^3-5^3+7^3-9^3+11^3-$.....N terms.

LANGUAGE USED: C

THEORY: WE WILL USE THE CONCEPT OF LOOPS HERE TO FIND THE SUM OF FOLLOWING SERIES.

CODE:

```
#include<stdio.h>

#include<math.h>

int main(void)

{

   int n,i,k=1;

   float sum=0;

   printf("Enter the number of terms");

   scanf("%d",&n);


   for(i=1; i<=n; i++)

     {

      sum= sum + (pow(-1,i)* pow(k,3));

      k=k+2;

     }

   printf("%f",sum);

}
```

OUTPUT:

```
Enter the number of terms5
-485.000000
```

29. Write a program to find the sum of following series: S = 1/1! + 2/2! + 3/3! + ... 7 terms.

LANGUAGE USED: C

THEORY: WE WILL USE THE CONCEPT OF LOOPS HERE TO FIND THE SUM OF FOLLOWING SERIES.

CODE:

```
#include<stdio.h>

int main()

{

   int i,j;

   float fact, sum=0.0 ;

   printf("sum of following series: S = 1/1! + 2/2! + 3/3! + upto. 7 terms \n");

  for (i=1; i<=7; i++)

  {

    fact=1;

    for(j=1;j<=i;j++)

    {

      fact= (fact*j);

    }

   sum= sum+ (i/fact);

  }

   printf("Sum of series= %f",sum);

}
```

OUTPUT:

```
sum of following series: S = 1/1! + 2/2! + 3/3! + upto. 7 terms
Sum of series= 2.718056
```

30. Write a program to convert binary number to decimal number.

Language used: C

Theory: user is supposed to enter the value of binary number, then dividing by 10 and taking remainder gives the values of digits of binary number and then multiplying by 2 power n gives the value of decimal number.

CODE:

```
#include<stdio.h>

#include<math.h>

int main(void)

{

   int n,b,i,deci=0;

   printf("Enter the binary number ");

   scanf("%d",&n);


   for(i=0; n>0; ++i)

   {

    b= n%10;

    deci= deci + b*pow(2,i);

    n= n/10;

   }

   printf("decimal equivalent= %d",deci);

}
```

OUTPUT:

```
Enter the binary number 1001
decimal equivalent= 9
```

31. Write a program to find the sum of following series: S = 1^4 + 3^4 + 5^4 + 7^4 + ... 100 terms

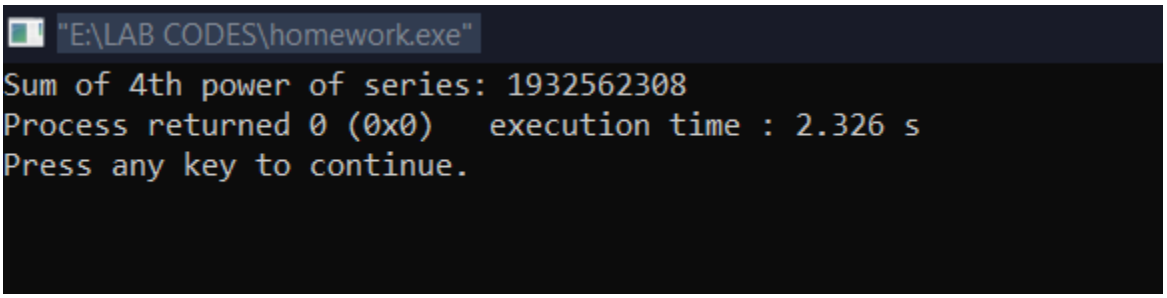
Language used- C

Theory: The concept of loops will be used here. Everytime when it executes the value of terms changes by 2 and the sum is stored in the sum variable with the forth power of k.

CODE:

```
#include<stdio.h>

#include<math.h>

int main()

{

 int k=1,i;

 unsigned long int sum=0;

 printf("Series: ");

 for(i=1;i<=100; i++)

 {

  printf(",%d",k);

  sum= sum+ pow(k,4);

  k=k+2;

 }

  printf("\nSum of 4th power of series: %d ",sum);

  return 0;

}
```

OUTPUT:



32. Write a program in C to print the following pattern:

* * *
* * *
* * *

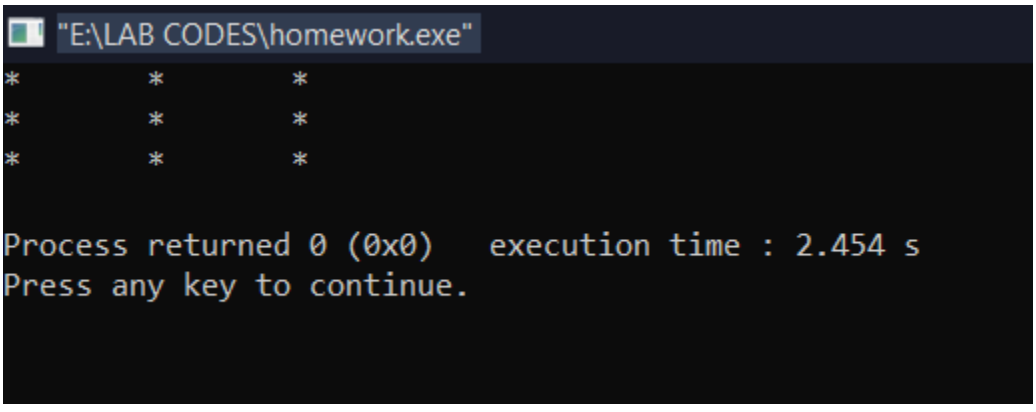
Language used: C

Theory: The concept of nested loops will be applied here in terms of i and j.

CODE:

```
#include<stdio.h>

int main(void)

{

   int i,j;

   for(i=1;i<=3;++i) {

      for(j=1;j<4;++j) {

         printf("*\t");

      }

      printf("\n");

   }

}
```

OUTPUT:



33. Write a program in C to print the following pattern:

```
1   2   3
1   2   3
12  3
```

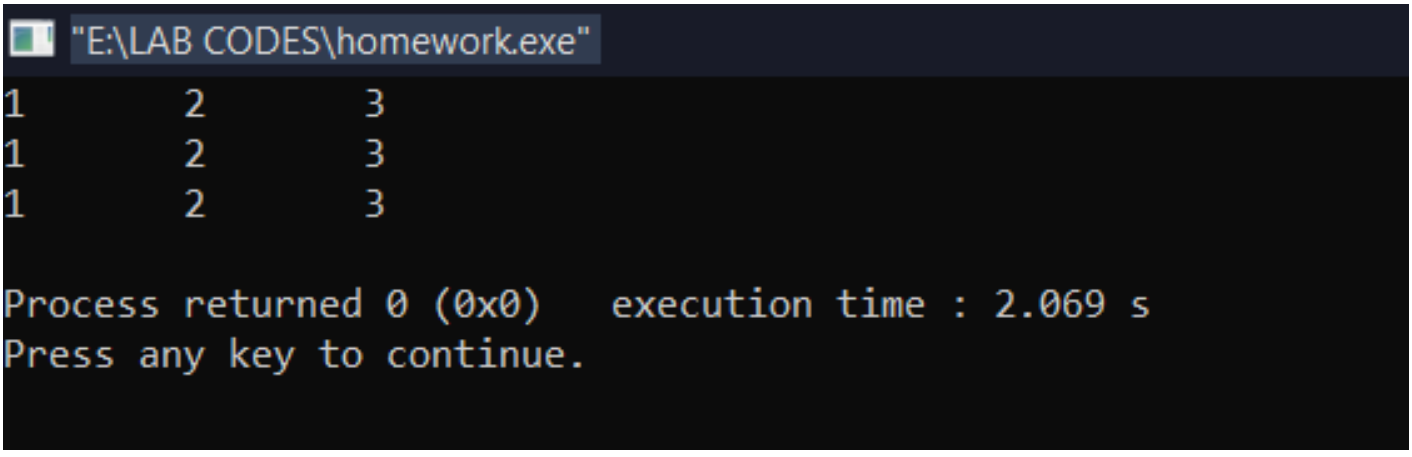
Language used: C

Theory: the concept of nested loops will be used here in terms of I and j. in this it is clear that the value of I and j will be equal.

CODE:

```
#include<stdio.h>

int main(void) {

   int i,j;

   for(i=1;i<=3;++i) {

      for(j=1;j<4;++j) {

         printf("%d\t",j);

      }

      printf("\n");

   }

}
```

OUTPUT:



34. Write a program in C to print the following pattern:

```
1   1   1
2   2   2
3   3   3
```

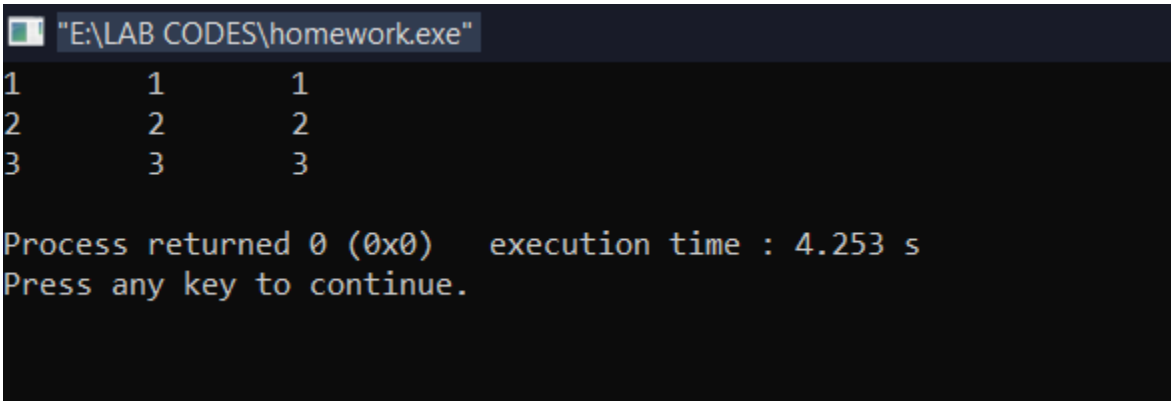
Language used: C

Theory: Nested loops will be used here. The value of I is printed here if we take I as outer loop.

CODE:

```
#include<stdio.h>

int main(void)

{

   int i,j;

   for(i=1;i<=3;++i)

   {

      for(j=1;j<4;++j)

      {

         printf("%d\t",i);

      }

      printf("\n");

   }

}
```

OUTPUT:



35. Write a program in C to print the following pattern:

```
3    2    1
3    2    1
3    2    1
```

Language used: C

Theory: The concept of nested loops will be used here, if we take outer loops in terms of I and inner in j then the value of j is printed.

CODE:

```
#include<stdio.h>

int main(void)

{

   int i,j;

   for(i=1;i<4;++i)

  {

      for(j=3;j>0;--j)

      {

         printf("%d\t",j);

      }

      printf("\n");

   }

}
```

OUTPUT:

```
"E:\LAB CODES\homework.exe"
3       2       1
3       2       1
3       2       1

Process returned 0 (0x0)   execution time : 1.157 s
Press any key to continue.
```

36. Write a program in C to print the following pattern:

```
3    3    3
2    2    2
1    1    1
```

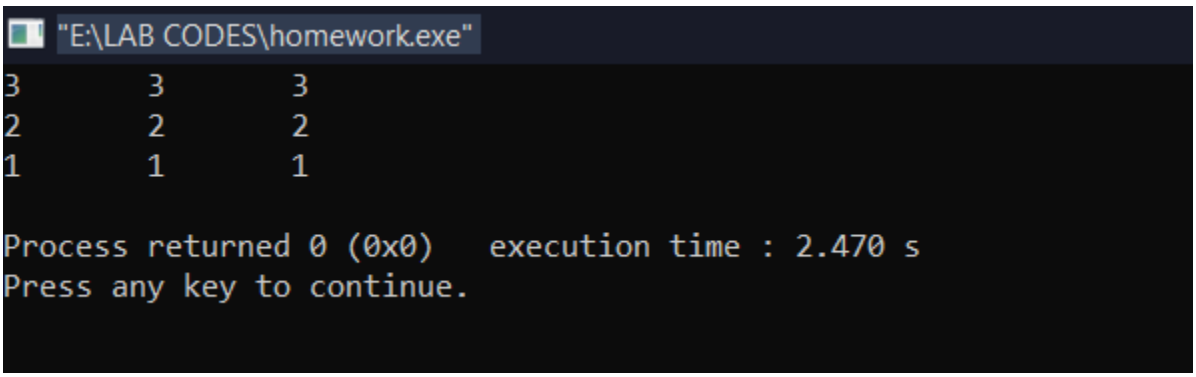
Language used: C

Theory: The concept of nested loops will be used here, if we take outer loops in terms of I and inner in j then the value of i is printed starting I with 3.

CODE:

```
#include<stdio.h>

int main(void)

{

 int i,j;

  for(i=3;i>0;--i)

  {

     for(j=3;j>0;--j)

    {

       printf("%d\t",i);

    }

    printf("\n");

  }

}
```

OUTPUT:



37. Write a program in C to print the following pattern:

```
*
*   *
*   *   *
```

Language used: C

Theory: The concept of nested loops will be used here, if we take outer loops in terms of I and inner in j but the j will be compared to I because the * is to be printed I times.

CODE:

```
#include<stdio.h>

int main(void)

{

   int i,j;

   for(i=1;i<=3;++i)

  {

      for(j=1;j<=i;++j)

      {

         printf("*\t");

      }

      printf("\n");

   }

}
```

OUTPUT:

```
"E:\LAB CODES\homework.exe"
*
*       *
*       *       *

Process returned 0 (0x0)   execution time : 2.457 s
Press any key to continue.
```

38. Write a program in C to print the following pattern:

```
1
1       2
1   2   3
```

Language used: C

Theory: The concept of nested loops will be used here, if we take outer loops in terms of I and inner in j but the j will be compared to I. and the numbers are to be printed j times.

CODE:

```
#include<stdio.h>

int main(void)

{

  int i,j;

   for(i=1;i<=3;++i)

  {

     for(j=1;j<=i;++j)

     {

        printf("%d\t",j);

     }

     printf("\n");

   }

}
```

OUTPUT:

```
"E:\LAB CODES\homework.exe"
1
1       2
1       2       3

Process returned 0 (0x0)   execution time : 0.981 s
Press any key to continue.
```

39. Write a program in C to print the following pattern:

```
1
2   2
3   3   3
```

Language used: C

Theory: The concept of nested loops will be used here, if we take outer loops in terms of I and inner in j but the j will be compared to I. and the numbers are to be printed i times.

CODE:

```
#include<stdio.h>

int main(void)

{

   int i,j;

   for(i=1;i<=3;++i)

  {

      for(j=1;j<=i;++j)

      {

         printf("%d\t",i);

      }

      printf("\n");

   }

}
```

OUTPUT:

```
"E:\LAB CODES\homework.exe"
1
2       2
3       3       3

Process returned 0 (0x0)   execution time : 0.855 s
Press any key to continue.
```

40. Write a program in C to print the following pattern:

```
3
3   2
3   2   1
```

Language used: C

Theory: the concept of nested loops will be used here, if we take outer loops in terms of I and inner in j but the j will be compared to I. but the value of I and j will start form 3 and the value of j will be printed.

CODE:

```
#include<stdio.h>

int main(void)

{

   int i,j;

   for(i=3;i>=1;i--)

  {

      for(j=3;j>=i;j--)

      {

         printf("%d\t",j);

      }

      printf("\n");

   }

}
```

OUTPUT:

```
"E:\LAB CODES\homework.exe"
3
3       2
3       2       1

Process returned 0 (0x0)   execution time : 3.855 s
Press any key to continue.
```

41. Write a program in C to print the following pattern:

```
3
2    2
1    1    1
```

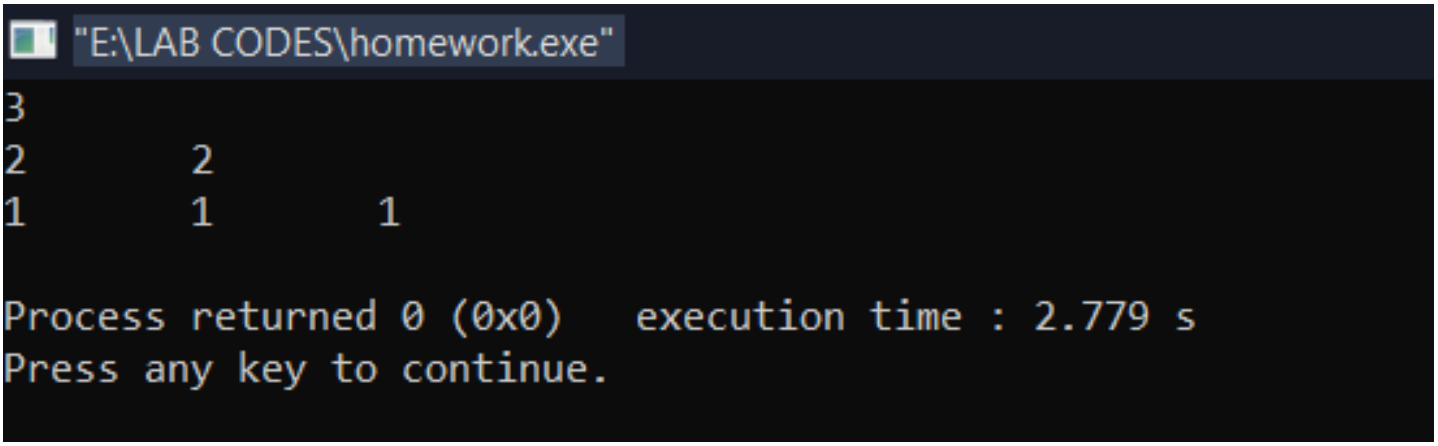
Language used: C

Theory: The concept of nested loops will be used here, if we take outer loops in terms of I and inner in j but the j will be compared to I. but the value of I and j will start form 3 and the value of i will be printed.

CODE:

```
#include<stdio.h>

int main(void)

{

   int i,j;

   for(i=3;i>=1;i--)

   {

      for(j=3;j>=i;j--)

      {

         printf("%d\t",i);

      }

      printf("\n");

   }

}
```

OUTPUT:



42. Write a program in C to print the following pattern:

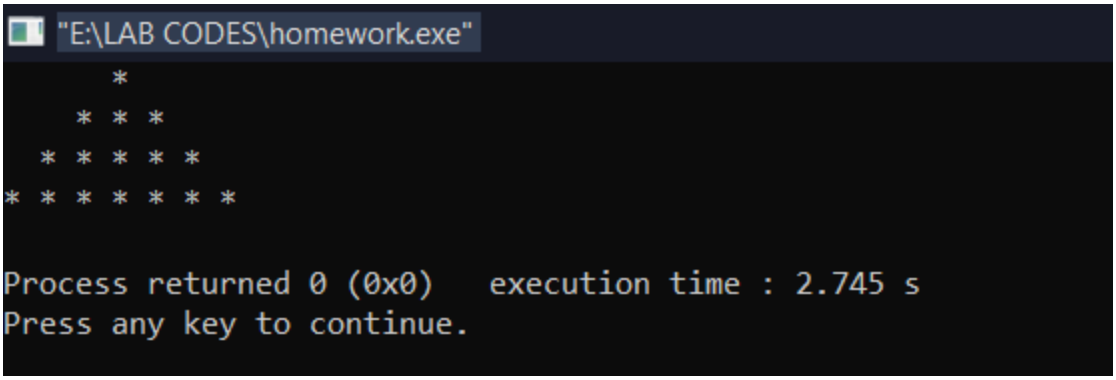
Language used: C

Theory: The concept of nested loops will be used here.

CODE:

```
#include <stdio.h>

int main()

{

  int i, space, rows=4 , k = 0;

  for (i = 1; i <= rows; ++i, k = 0)

  {

    for (space = 1; space <= rows - i; ++space)

    {

      printf("  ");

    }

    while (k != 2 * i - 1)

    {

      printf("* ");

      ++k;

    }

    printf("\n");

  }

  return 0;

}
```

OUTPUT:


"E:\LAB CODES\homework.exe"
*
* * *
* * * * *
* * * * * * *
Process returned 0 (0x0) execution time : 2.745 s
Press any key to continue.

43. Write a program in C to print the following pattern:

```
        1
      1 2 1
    1 2 3 2 1
  1 2 3 4 3 2 1
```

Language used: C

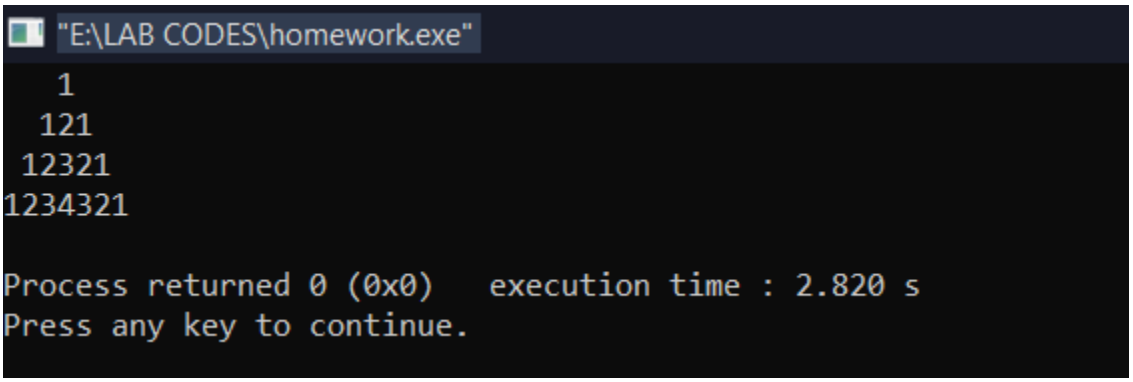
Theory: The concept of nested loops will be used here.

CODE:

```
#include<stdio.h>

int main()

{int i,j,n=4,k;

for(i=1;i<=n;i++)

 {

  for(j=1;j<=n-i;j++)

   printf(" ");

   {

    for(j=1;j<=i;j++)

    printf("%d",j);

    j=1;

    for(j=i;i>=j;j--)

    if(j==0)

    break;

    else

    {

     k=j-1;

     if(k==0)

     break;

     else

     printf("%d",k);
```

```
   }
    printf("\n");
   }
  }
 return 0 ;
}
```

OUTPUT:



44. Write a program in C to print the following pattern:

```
                5
            5   4   5
        5   4   3   4   5
    5   4   3   2   3   4   5
5   4   3   2   1   2   3   4   5
```

LANGUAGE USED: C

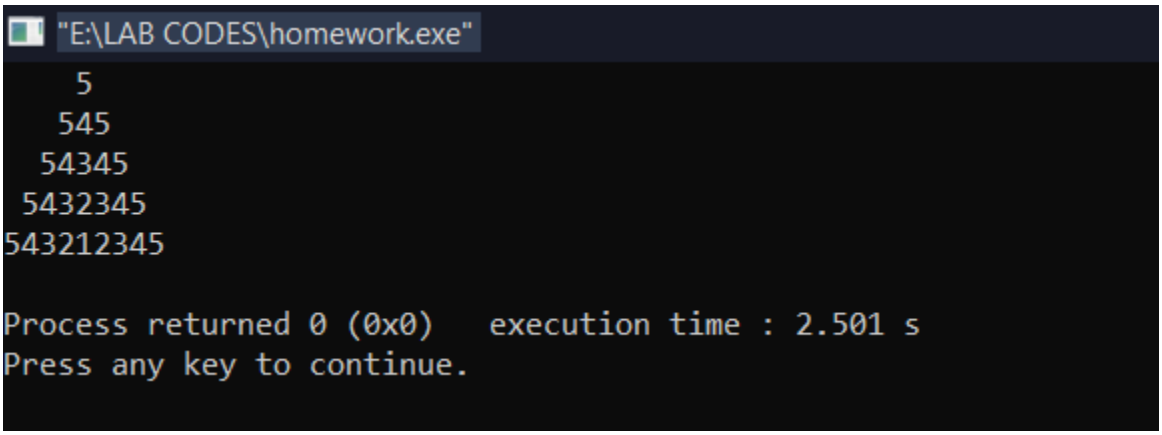
THEORY: THE CONCEPT OF NESTED LOOPS WILL BE USED HERE.

CODE:

```
#include<stdio.h>


int main()

{

   int i,j,k,l;

   for(i=5;i>=1;i--)

   {

      for(j=1;j<i;j++)

         printf(" ");

      for(k=5;k>=i;k--)

         printf("%d",k);

      if(i==5)

      {


      }

         else

         {

            k=k+2;

            for(l=k;l<=5;l++)

               printf("%d",l);
```

```
        }
      printf("\n");
   }
}
```

OUTPUT:



45. Write a program in C to print the following pattern:

```
1
0 1
1 0 1
0 1 0 1
1 0 1 0 1
```

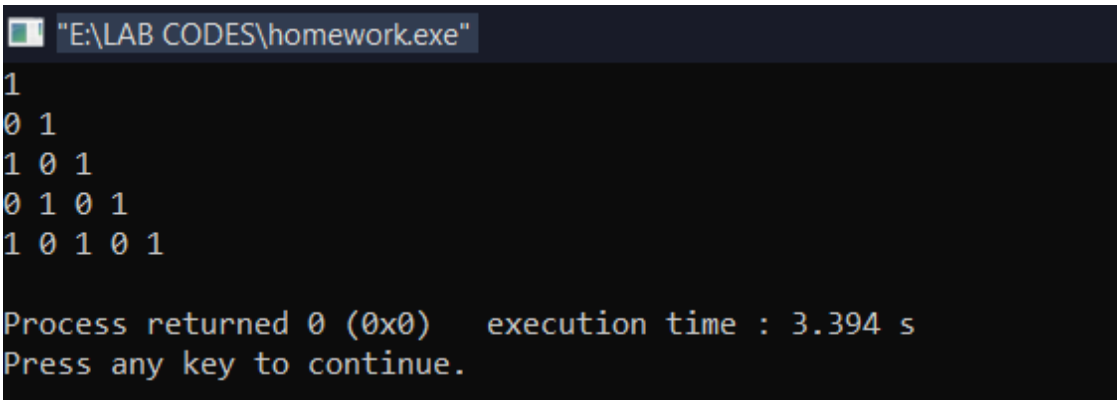
LANGUAGE USED: C

THEORY: THE CONCEPT OF NESTED LOOPS WILL BE USED HERE.

CODE:

```
#include<stdio.h>

int main()

{

 int i, j, rows=5 ;

 int count = 1;

for (i = 0; i < rows; i++)

 {

  for (j = 0; j <= i; j++)

   {

    printf("%d ", count);

    count = !count;

   }

  count = i % 2;

  printf("\n");

 }

 return(0);

}
```

OUTPUT:


"E:\LAB CODES\homework.exe"
1
0 1
1 0 1
0 1 0 1
1 0 1 0 1
Process returned 0 (0x0) execution time : 3.394 s
Press any key to continue.

46. Write a program to print all prime numbers <= a given number.

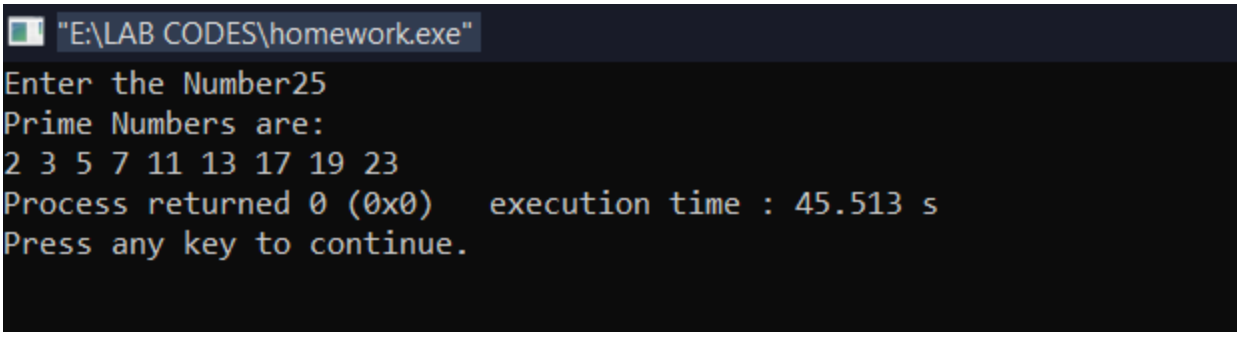
Language used: C

Theory: Iterate from 2 to N, and check for prime. If it is a prime number, print the number.

CODE:

```
#include<stdio.h>

int main()

{

   int n,i,fact,j;

   printf("Enter the Number");

   scanf("%d",&n);

   printf("Prime Numbers are: \n");

   for(i=1; i<=n; i++)

   {

      fact=0;

      for(j=1; j<=n; j++)

      {

         if(i%j==0)

            fact++;

      }

      if(fact==2)

         printf("%d " ,i);

   }

   return 0;

}
```

OUTPUT:



47. Write a program to convert Decimal no to Binary No.

Language used: C

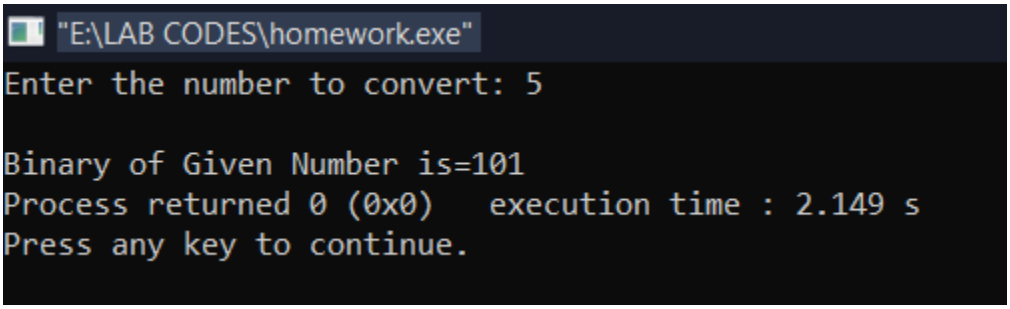
Theory: Store the remainder when the number is divided by 2 in an array.
Divide the number by 2
Repeat the above two steps until the number is greater than zero.
Print the array in reverse order now.

CODE:

```
#include<stdio.h>

int main()

{

 int a[10],n,i;

 printf("Enter the number to convert: ");

 scanf("%d",&n);

 for(i=0;n>0;i++)

 {

  a[i]=n%2;

  n=n/2;

 }

 printf("\nBinary of Given Number is=");

 for(i=i-1;i>=0;i--)

 {

  printf("%d",a[i]);

 }

 return 0;
```

}

OUTPUT:



48. Write a program to find product, sum, average, max and min from a list of n numbers.

LANGUAGE USED: C

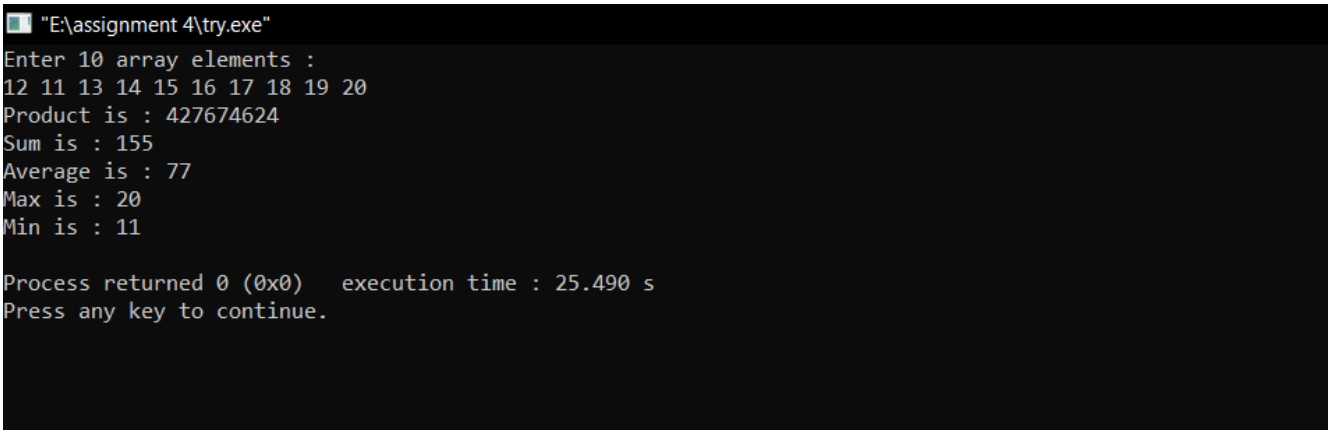
THEORY: HERE WE WILL FIND THE PRODUCT, SUM, AVERAGE, MAX. AND MIN. ARRAY ELEMENT WITH THEIR POSITION.

CODE:

```
#include<stdio.h>

int main()

{       int a[10],product=1,sum=0;

    •   printf("Enter 10 array elements : \n");
    •   for(int i=0;i<10;i++)
    •   {
    •      scanf("%d",&a[i]);
    •   }
    •   for(int i=0;i<10;i++)
    •   {
    •      product = product * a[i];
    •   }
    •   printf("Product is : %d\n",product);
    •   for(int i=0;i<10;i++)
    •   {
    •      sum = sum + a[i];
    •   }
    •   printf("Sum is : %d\n",sum);
    •   int average=sum/2;
    •   printf("Average is : %d\n",average);
```

```
    int max = a[0];
    for(int i=0;i<10;i++)
    {
       if(a[i]>max)
          max = a[i];
    }
    printf("Max is : %d\n",max);
    int min = a[0];
    for(int i=0;i<10;i++)
    {
       if(a[i]<min)
          min = a[i];
    }
    printf("Min is : %d\n",min);
    return 0;
}
```

OUTPUT:



49. Write a program in C to display the index of smallest and largest element in 10 integers.

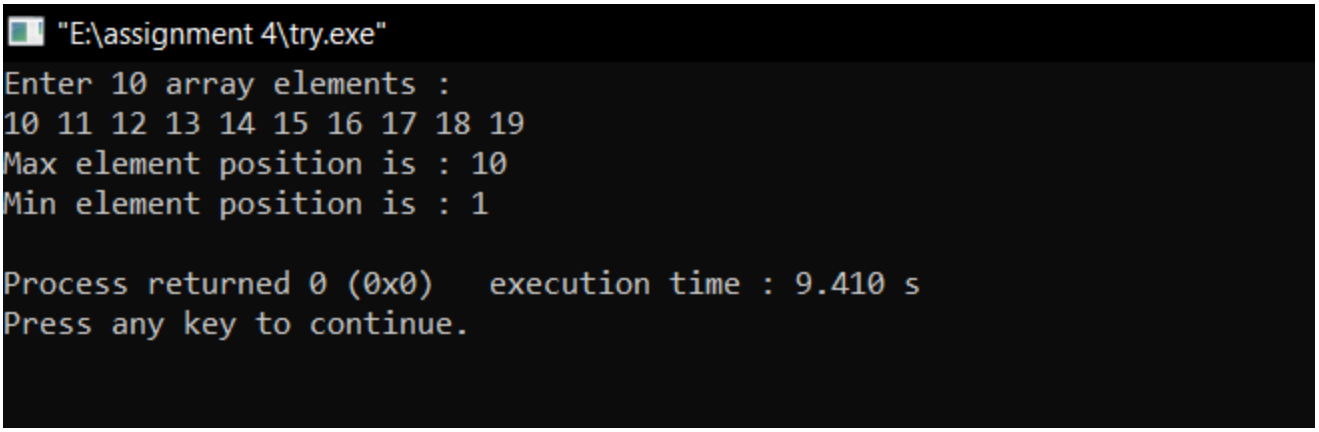
LANGUAGE USED: C

THEORY: HERE WE WILL FIND THE SMALLEST AND LARGEST ARRAY ELEMENT WITH THEIR POSITION.

CODE:

```
#include<stdio.h>
int main()
{
   int a[10];
   printf("Enter 10 array elements : \n");
   for(int i=0;i<10;i++)
   {
      scanf("%d",&a[i]);
   }
   int temp=0;
   for(int i=1;i<10;i++)
   {
      if(a[i]>a[temp])
         temp=i;

   }
   printf("Max element position is : %d\n",temp);
   int temp2=0;
   for(int i=1;i<10;i++)
   {
      if(a[i]<a[temp2])
         temp2=i;
   }
   printf("Min element position is : %d\n",temp2);
   return 0;
}
```

OUTPUT:


"E:\assignment 4\try.exe"
Enter 10 array elements :
10 11 12 13 14 15 16 17 18 19
Max element position is : 10
Min element position is : 1
Process returned 0 (0x0) execution time : 9.410 s
Press any key to continue.

50. Write a program in C to display the index of smallest and largest element in 3 X 4 matrix of integers.

LANGUAGE USED: C

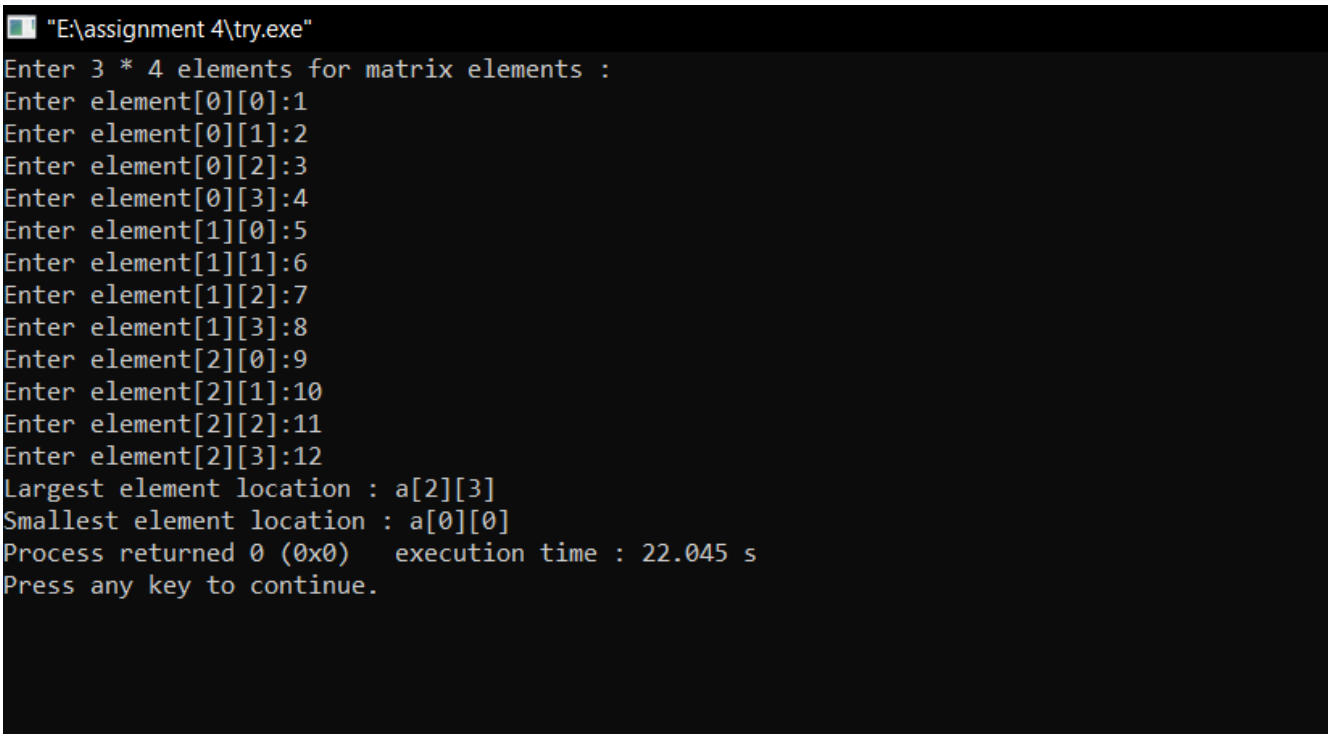
THEORY: HERE WE WILL FIND THE SMALLEST AND LARGEST ARRAY ELEMENT WITH THEIR POSITION.

CODE:

```
#include<stdio.h>
int main()
{
   int a[3][4]={0};
   int brow=0,bcol=0,srow=0,scol=0;
   printf("Enter 3 * 4 elements for matrix elements :\n");
   for(int i=0;i<3;i++)
   {
      for(int j=0;j<4;j++)
      {  printf("Enter element[%d][%d]:",i,j);
         scanf("%d",&a[i][j]);
      }
   }
   int big=a[0][0];
   int small=a[0][0];
   for(int i=0;i<3;i++)
   {
      for(int j=0;j<4;j++)
      {
         if(a[i][j]>big)
         {
            big = a[i][j];
            brow=i;
            bcol=j;
         }
         if(a[i][j]<small)
         {
            small = a[i][j];
            srow=i;
            scol=j;
         }
      }
   }
   printf("Largest element location : a[%d][%d]\n",brow,bcol);
```

- printf("Smallest element location : a[%d][%d]",srow,scol);
- return 0;
- }

OUTPUT:



51. Write a program in C that accepts N*N matrix as input and print transpose of this matrix .

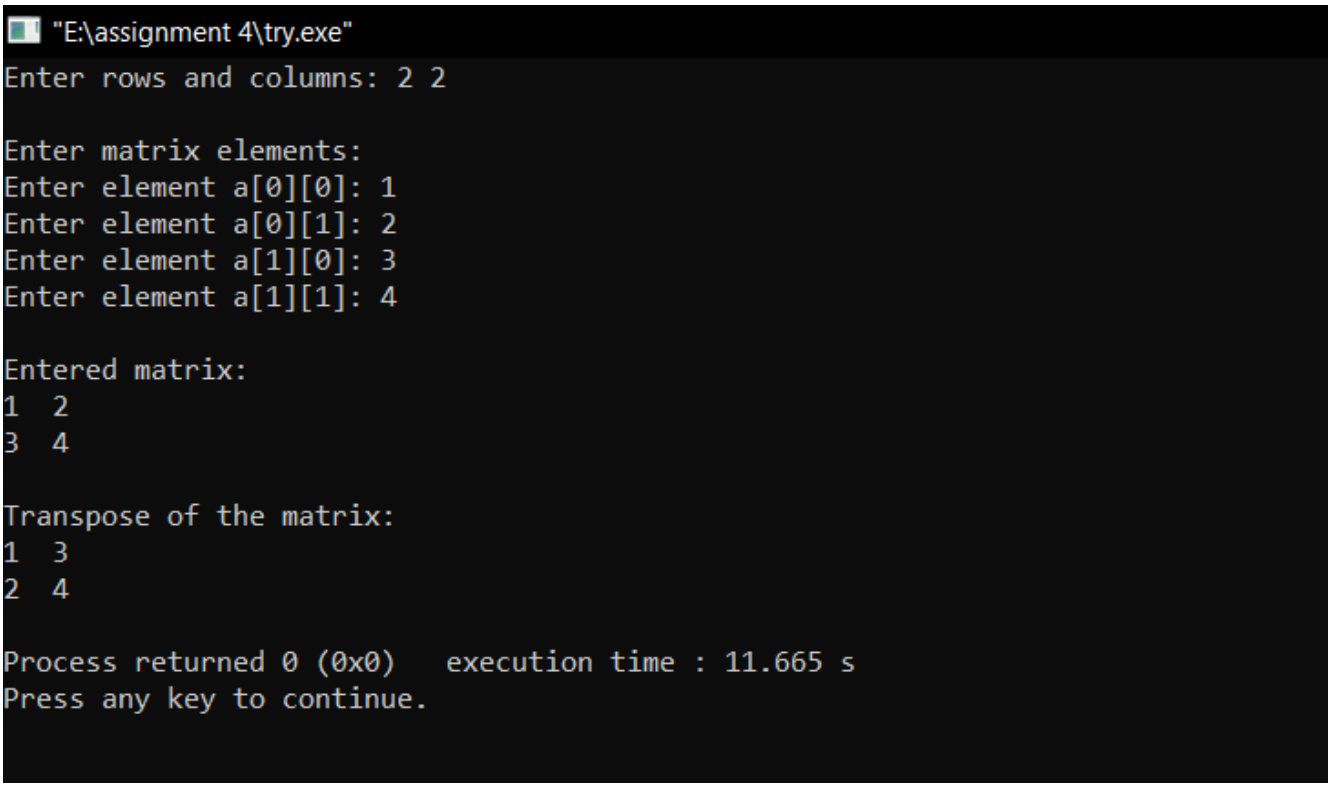
LANGUAGE USED: C

THEORY: THE TRANSPOSE OF A MATRIX IS A NEW MATRIX THAT IS OBTAINED BY EXCHANGING THE ROWS AND COLUMNS.

CODE:

```
#include<stdio.h>
int main()
{
   int a[10][10], transpose[10][10], r, c, i, j;
   printf("Enter rows and columns: ");
   scanf("%d %d", &r, &c);
   printf("\nEnter matrix elements:\n");
   for (i = 0; i < r; ++i)
      for (j = 0; j < c; ++j) {
         printf("Enter element a[%d][%d]: ", i, j);
         scanf("%d", &a[i][j]);
      }
   printf("\nEntered matrix: \n");
   for (i = 0; i < r; ++i)
      for (j = 0; j < c; ++j) {
         printf("%d  ", a[i][j]);
         if (j == c - 1)
            printf("\n");
      }
   for (i = 0; i < r; ++i)
      for (j = 0; j < c; ++j) {
         transpose[j][i] = a[i][j];
      }
   printf("\nTranspose of the matrix:\n");
   for (i = 0; i < c; ++i)
      for (j = 0; j < r; ++j) {
         printf("%d  ", transpose[i][j]);
         if (j == r - 1)
            printf("\n");
      }
   return 0;
}
```

OUTPUT:


"E:\assignment 4\try.exe"
Enter rows and columns: 2 2
Enter matrix elements:
Enter element a[0][0]: 1
Enter element a[0][1]: 2
Enter element a[1][0]: 3
Enter element a[1][1]: 4
Entered matrix:
1 2
3 4
Transpose of the matrix:
1 3
2 4
Process returned 0 (0x0) execution time : 11.665 s
Press any key to continue.

52. Write a program to accept two matrices of some order. (Order must be given by user) find out the sum of these matrices and print the sum of matrices.

LANGUAGE USED: C

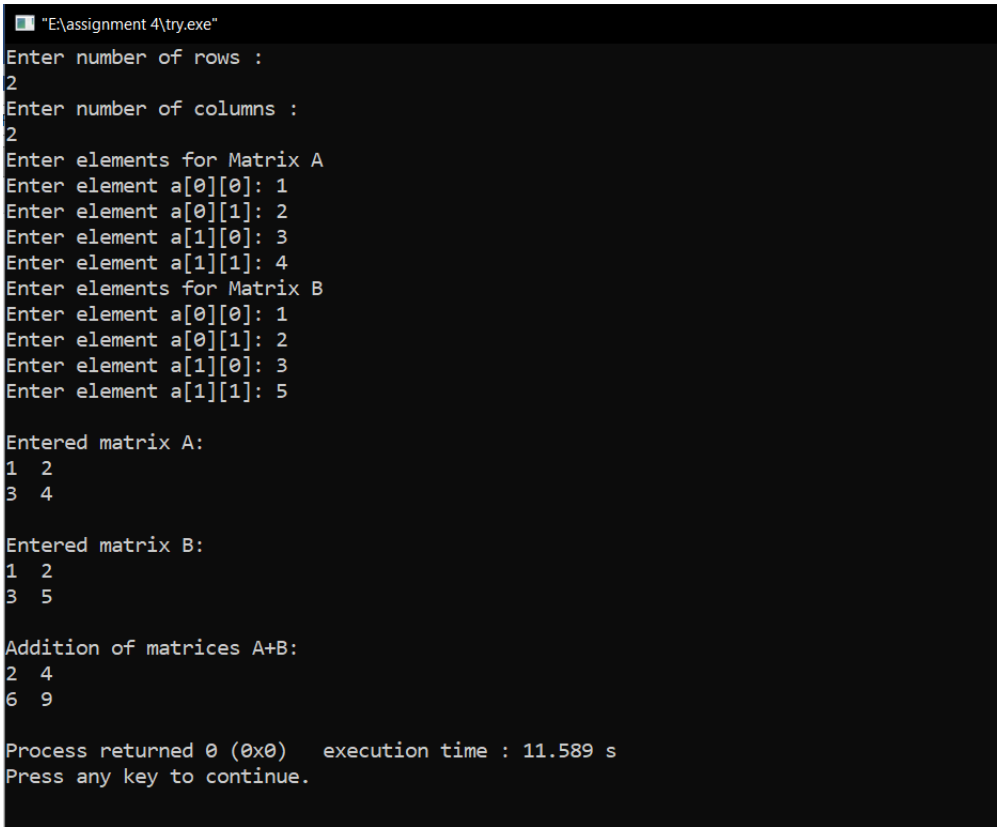
THEORY: IN THIS PROGRAM, WE NEED TO SUM TWO MATRICES AND PRINT THE RESULTING MATRIX.

CODE:

```
#include<stdio.h>
int main()
{
   int n,m;
   printf("Enter number of rows : \n");
   scanf("%d",&n);
   printf("Enter number of columns : \n");
   scanf("%d",&m);
   int a[n][m],b[n][m];
   printf("Enter elements for Matrix A\n");
   for (int i = 0; i < n; ++i)
   {
      for (int j = 0; j < m; ++j)
      {
         printf("Enter element a[%d][%d]: ", i, j);
         scanf("%d", &a[i][j]);
      }
   }
   printf("Enter elements for Matrix B\n");
   for (int i = 0; i < n; ++i)
   {
      for (int j = 0; j < m; ++j)
      {
         printf("Enter element a[%d][%d]: ", i, j);
         scanf("%d", &b[i][j]);
      }
   }
   printf("\nEntered matrix A: \n");
   for (int i = 0; i < n; ++i)
      for (int j = 0; j < m; ++j) {
         printf("%d  ", a[i][j]);
         if (j == m - 1)
            printf("\n");
```

```
        }
    printf("\nEntered matrix B: \n");
    for (int i = 0; i < n; ++i)
        for (int j = 0; j < m; ++j) {
            printf("%d  ", b[i][j]);
            if (j == m - 1)
                printf("\n");
        }
    printf("\nAddition of matrices A+B: \n");
    for (int i = 0; i < n; ++i)
        for (int j = 0; j < m; ++j) {
            printf("%d  ", a[i][j] + b[i][j]);
            if (j == m - 1)
                printf("\n");
        }

    return 0;
}
```

OUTPUT:



53. Write a program to find out the product / Multiplication of two matrices and print the product matrix. (order of matrices must be given by user)

LANGUAGE USED: C

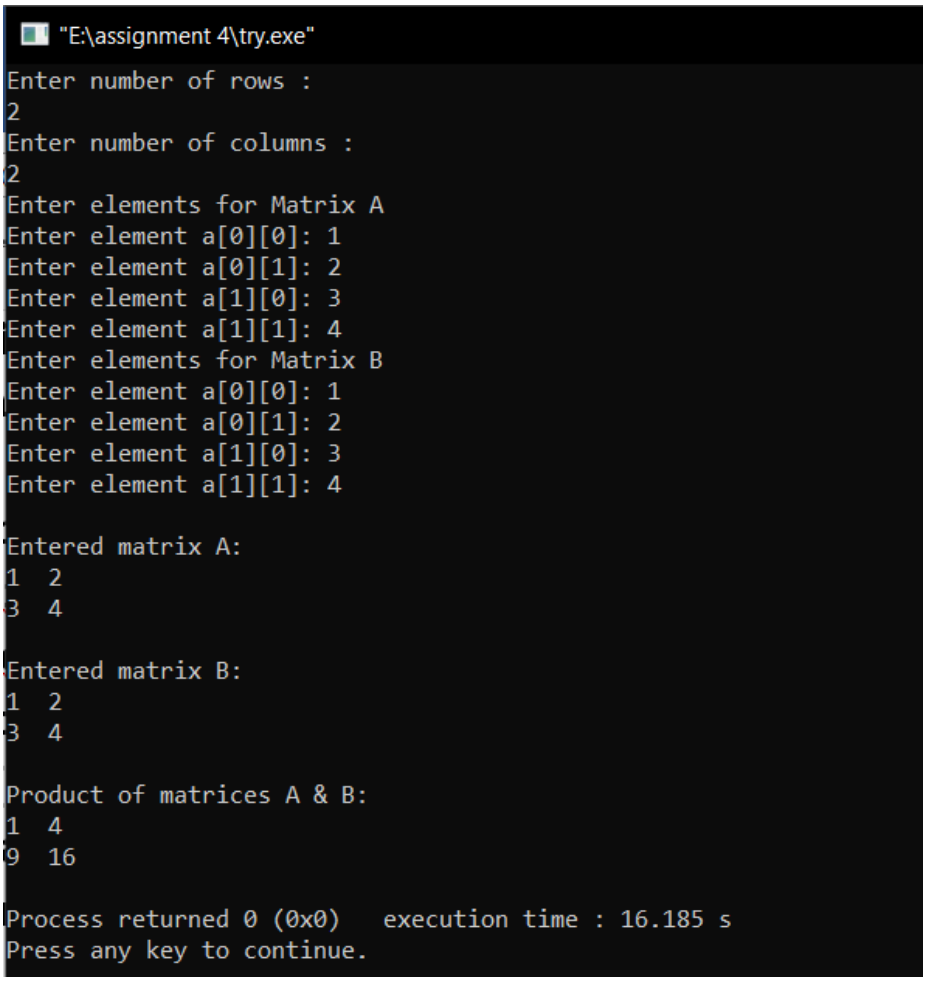
THEORY: IN THIS PROGRAM, WE NEED TO MULTIPLY TWO MATRICES AND PRINT THE RESULTING MATRIX.

CODE:

```
#include<stdio.h>
int main()
{
   int n,m;
   printf("Enter number of rows : \n");
   scanf("%d",&n);
   printf("Enter number of columns : \n");
   scanf("%d",&m);
   int a[n][m],b[n][m];
   printf("Enter elements for Matrix A\n");
   for (int i = 0; i < n; ++i)
   {
      for (int j = 0; j < m; ++j)
      {
         printf("Enter element a[%d][%d]: ", i, j);
         scanf("%d", &a[i][j]);
      }
   }
   printf("Enter elements for Matrix B\n");
   for (int i = 0; i < n; ++i)
   {
      for (int j = 0; j < m; ++j)
      {
         printf("Enter element a[%d][%d]: ", i, j);
         scanf("%d", &b[i][j]);
      }
   }
   printf("\nEntered matrix A: \n");
   for (int i = 0; i < n; ++i)
      for (int j = 0; j < m; ++j) {
         printf("%d  ", a[i][j]);
         if (j == m - 1)
            printf("\n");
```

```
        }
    printf("\nEntered matrix B: \n");
    for (int i = 0; i < n; ++i)
        for (int j = 0; j < m; ++j) {
            printf("%d  ", b[i][j]);
            if (j == m - 1)
                printf("\n");
        }
    printf("\nProduct of matrices A & B: \n");
    for (int i = 0; i < n; ++i)
        for (int j = 0; j < m; ++j) {
            printf("%d  ", a[i][j] * b[i][j]);
            if (j == m - 1)
                printf("\n");
        }

    return 0;
}
```

OUTPUT:



54. Write a program to accept two matrices of some order. (Order must be given by user) find out the subtraction of these matrices and print the sum of matrices.

LANGUAGE USED: C

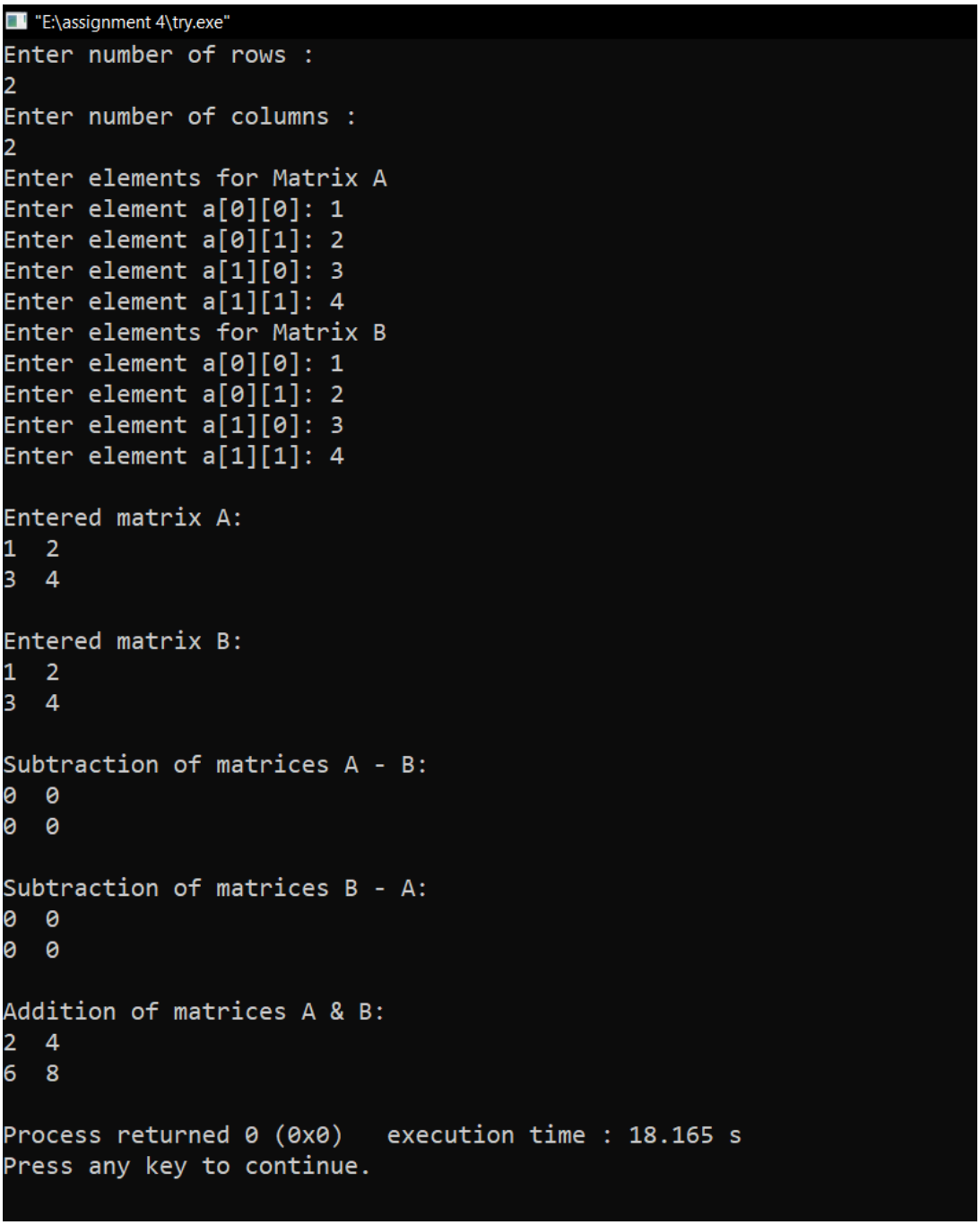
THEORY: IN THIS PROGRAM, WE NEED TO ADD AND SUBTRACT TWO MATRICES AND PRINT THE RESULTING MATRIX.

CODE:

```
#include<stdio.h>
int main()
{
   int n,m;
   printf("Enter number of rows : \n");
   scanf("%d",&n);
   printf("Enter number of columns : \n");
   scanf("%d",&m);
   int a[n][m],b[n][m];
   printf("Enter elements for Matrix A\n");
   for (int i = 0; i < n; ++i)
   {
      for (int j = 0; j < m; ++j)
      {
         printf("Enter element a[%d][%d]: ", i, j);
         scanf("%d", &a[i][j]);
      }
   }
   printf("Enter elements for Matrix B\n");
   for (int i = 0; i < n; ++i)
   {
      for (int j = 0; j < m; ++j)
      {
         printf("Enter element a[%d][%d]: ", i, j);
         scanf("%d", &b[i][j]);
      }
   }
   printf("\nEntered matrix A: \n");
   for (int i = 0; i < n; ++i)
      for (int j = 0; j < m; ++j) {
         printf("%d  ", a[i][j]);
         if (j == m - 1)
```

```
            printf("\n");
        }
    printf("\nEntered matrix B: \n");
    for (int i = 0; i < n; ++i)
        for (int j = 0; j < m; ++j) {
            printf("%d  ", b[i][j]);
            if (j == m - 1)
                printf("\n");
        }
    printf("\nSubtraction of matrices A - B: \n");
    for (int i = 0; i < n; ++i)
        for (int j = 0; j < m; ++j) {
            printf("%d  ", a[i][j] - b[i][j]);
            if (j == m - 1)
                printf("\n");
        }
        printf("\nSubtraction of matrices B - A: \n");
    for (int i = 0; i < n; ++i)
        for (int j = 0; j < m; ++j) {
            printf("%d  ", b[i][j] - a[i][j]);
            if (j == m - 1)
                printf("\n");
        }
    printf("\nAddition of matrices A & B: \n");
    for (int i = 0; i < n; ++i)
        for (int j = 0; j < m; ++j) {
            printf("%d  ", a[i][j] + b[i][j]);
            if (j == m - 1)
                printf("\n");
        }

    return 0;
}
```

OUTPUT:


"E:\assignment 4\try.exe"
Enter number of rows :
2
Enter number of columns :
2
Enter elements for Matrix A
Enter element a[0][0]: 1
Enter element a[0][1]: 2
Enter element a[1][0]: 3
Enter element a[1][1]: 4
Enter elements for Matrix B
Enter element a[0][0]: 1
Enter element a[0][1]: 2
Enter element a[1][0]: 3
Enter element a[1][1]: 4
Entered matrix A:
1 2
3 4
Entered matrix B:
1 2
3 4
Subtraction of matrices A - B:
0 0
0 0
Subtraction of matrices B - A:
0 0
0 0
Addition of matrices A & B:
2 4
6 8
Process returned 0 (0x0) execution time : 18.165 s
Press any key to continue.

55. Write a C Program to implement Simple Calculator (Addition, Subtraction, Multiplication, Division) using the concept of function

LANGUAGE USED: C

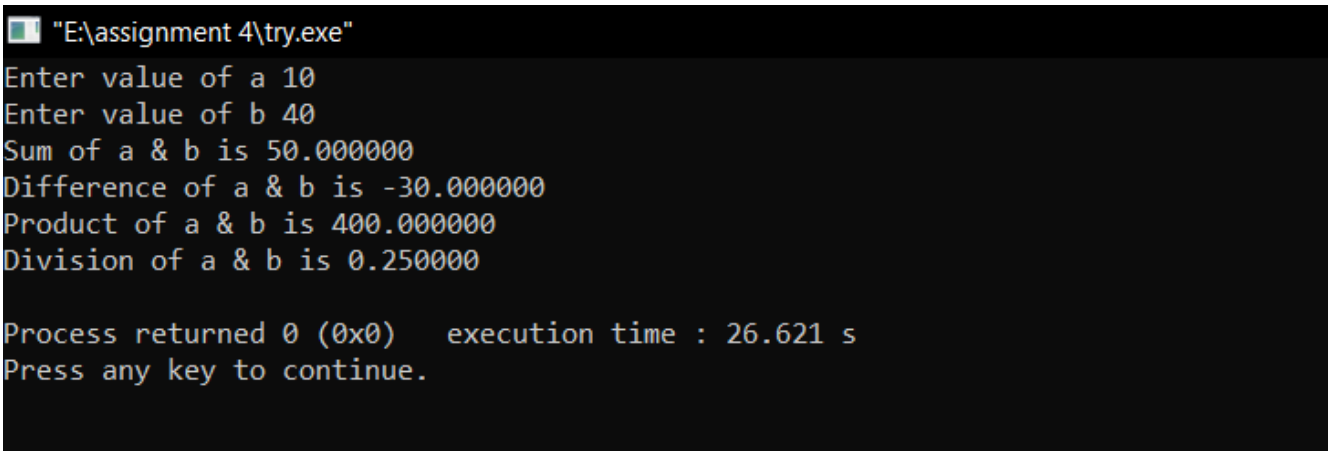
THEORY: THIS PROGRAM PROMPTS USER FOR ENTERING ANY INTEGER NUMBER, FINDS THE ADDITION, SUBTRACTION, MULTIPLICATION, DIVISION OF INPUT NUMBER AND DISPLAYS THE OUTPUT ON SCREEN. WE WILL USE A USER DEFINED FUNCTIONS TO PERFORM THE TASK.

CODE:

```
#include<stdio.h>
float sum(float a, float b)
{
   return a+b;
}
float diff(float a, float b)
{
   return a-b;
}
float prod(float a, float b)
{
   return a*b;
}
float div(float a, float b)
{
   if(b==0)
      return 0;
   return a/b;
}
int main()
{
   float a ,b;
   printf("Enter value of a ");
   scanf("%f",&a);
   printf("Enter value of b ");
   scanf("%f",&b);
   float c;
   c=sum(a,b);
   printf("Sum of a & b is %f\n",c);
   c=diff(a,b);
   printf("Difference of a & b is %f\n",c);
   c=prod(a,b);
```

```
    printf("Product of a & b is %f\n",c);
    c=div(a,b);
    printf("Division of a & b is %f\n",c);
    return 0;
}
```

OUTPUT:



56. Write a C Program to swap two values using function

Language used: C

Theory: Swapping two variables refers to mutually exchanging the values of the variables. Generally, this is done with the data in memory.

CODE:

```
#include<stdio.h>
int swap(int a, int b)
{
   int temp = a;
   a=b;
   b=temp;
   printf("Swapped numbers are   :%d %d",a,b);
}
int main()
{
   int a ,b;
   printf("Enter value of a and b :");
   scanf("%d %d",&a, &b);
   swap(a,b);
   return 0;
}
```

OUTPUT:

```
"E:\assignment 4\try.exe"
Enter value of a and b :10 23
Swapped numbers are    :23 10
Process returned 0 (0x0)   execution time : 3.874 s
Press any key to continue.
```

57. Write a C Program to Calculate the factorial of a number using function

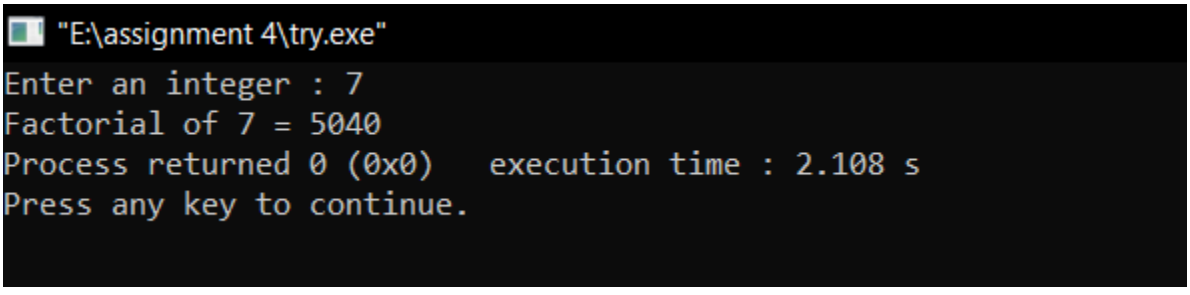
Language used: C

Theory: This Program prompts user for entering any integer number, finds the factorial of input number and displays the output on screen. We will use a user defined function to perform the task.

CODE:

```
#include<stdio.h>
void fact(int n)
{  unsigned long long fact=1;
   if(n<0)
   {
      printf("Factorial of negative numbers doesn't exist.");
   }
   else
   {
      for(int i=1;i<=n;i++)
      {
         fact*=i;
      }
      printf("Factorial of %d = %llu", n, fact);
   }

}
int main()
{
   int n;
   printf("Enter an integer : ");
   scanf("%d",&n);
   fact(n);
   return 0;
}
```

OUTPUT:



58. Write a C Program to Calculate the factorial of a number using recursion

Language used: C

Theory: This Program prompts user for entering any integer number, finds the factorial of input number and displays the output on screen. We will use a recursive user defined function to perform the task. Here we have a function fact that calls itself in a recursive manner to find out the factorial of input number.

CODE:

```
#include<stdio.h>
long int fact(int n);
int main()
{
   int n;
   printf("Enter an integer : ");
   scanf("%d",&n);
   int x = fact(n);
   printf("Factorial of %d = %ld", n, x);
   return 0;
}
long int fact(int n)
{
   if(n>=1)
      return n*fact(n-1);
   else
      return 1;
}
```

OUTPUT: