|  |
| --- |
|  |
| **LAKSHIT SHARMA**  **CSE 4X** |
|  |

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

(b) Write a C program to add three numbers

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

(b) Write a C program to calculate simple interest

1. 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
2. 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
3. (a) Write a C program to swap two variables using a third variable

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

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

(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

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

1. Write a C program to check whether a given year is Leap year or not
2. (a) Write a C program to check whether a triangle is Equilateral, scalene, or isosceles

(b) Write a C program to check whether a triangle is right angles, obtuse, acute triangle

1. Write a C program to covert temperature from Fahrenheit to Celsius and Celsius to Fahrenheit (User must provide the choice of type of temperature)
2. (a) Write a C program to check whether a character is an alphabet, digit

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

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

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

1. Write a program in C to implement Simple Calculator.
2. WAP to calculate the root of a Quadratic Equation
3. WAP to to accept a coordinate point in a XY coordinate system and determine in which quadrant the coordinate point lies.
4. 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.
5. 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/-

1. 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.
2. Write a program to find the factorial of any number.
3. WAP to calculate the root of a Quadratic Equation
4. Write a program to print Fibonacci sequence
5. 0 1 1 2 3 5 8 13…… N terms and prints the sum of sequence.
6. Write a program in C to accept an integer numbers and find sum of digits.
7. Write a program in C to accept an integer numbers and find reverse of this number and check this number for palindrome
8. Write a program in C to accept an integer numbers and to check a number is Armstrong or not
9. Write a program in C to accept an integer numbers and to check a number is Perfect or not
10. Write a program to find the sum of following series: S = 2+4+6+8+……………N terms.
11. Write a program to check a number whether it is prime number or not.
12. Write a program to find the sum of following series:

1 – 1/2 + 1/3 – 1/4 + 1/5 - …… up to n terms.

1. Write a program to find the sum of following series:

1! + 2! + 3! + 4! + ….. + n!

1. Write a program to find the sum of following series:

S = -1 3 + 33 - 5 3 + 73 – 9 3 + 113 - ……..N terms.

1. Write a program to find the sum of following series:

S = 1/1! + 2/2! + 3/3! + ………….. 7 terms.

1. Write a program to convert binary number to decimal number.
2. Write a program to find the sum of following series:

S = 14 + 34 + 54  + 74 + …………….. 100 terms

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

\* \* \*

\* \* \*

\* \* \*

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

1 2 3

1 2 3

1 2 3

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

1 1 1

2 2 2

3 3 3

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

3 2 1

3 2 1

3 2 1

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

3 3 3

2 2 2

1 1 1

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

\*

\* \*

\* \* \*

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

1

1. 2

1 2 3

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

1

2 2

3 3 3

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

3

3 2

3 2 1

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

3

2 2

1 1 1

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

\*

\* \* \*

\* \* \* \* \*

\* \* \* \* \* \* \*

1. 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

1. 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

1. 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

1. Write a program to print all prime numbers <= a given number.
2. Write a program to convert Decimal no to Binary No.
3. Write a program to find product, sum, average, max and min from a list of n numbers.
4. Write a program in C that accepts N\*N matrix as input and print transpose of this matrix
5. 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.
6. 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)
7. 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.
8. Write a C Program to implement Simple Calculator (Addition, Subtraction, Multiplication, Division) using the concept of function
9. Write a C Program to swap two values using function
10. Write a C Program to Calculate the factorial of a number using function
11. Write a C Program to Calculate the factorial of a number using recursion
12. Write a C program to check whether a number is even or odd using functions.
13. Write a C program to check whether a number is Prime, Armstrong or perfect number using functions.
14. Write a C program to find all prime numbers between given interval using functions.
15. Write a C program to print all strong numbers between given interval using functions.
16. Write a C program to find power of any number using recursion
17. Declare a structure name student containing members name, roll\_no, marks. Create an array of 30 such students. Write a program to read and display the contents of array.
18. Write a simple database program in C which stores personal details of 100 persons such as Name, Date of Birth, Address, Phone number etc.
19. Write a program in ‘C’ that compares two given dates. To store a date, use a struct that contains three members namely day, month, and year. If the dates are equal, then display message as “equal” otherwise “Unequal”.
20. Write a program which reads your name from the keyboard and outputs a list of ASCII codes, which represent your name.
21. Write a program which will read a text and count all occurrences of all characters which are part of text.
22. Write a program which will read a text and count all occurrences of a particular word.
23. Write a program which reads a string from the keyboard and determines whether the string is a palindrome (Ignore Capitalization)
24. Write macro definition with arguments for calculation of simple interest and amount. Store these macro definitions in a file called ‘Interest.h”. Include this file in your program and use the macro definition for calculating simple interest and amount.
25. Write a program to copy the contents of one file to another file.
26. Write a program which will store ten integers to one file and squares of these numbers to another file.
27. Write a program which will store ten integers to one file and stores the odd and even numbers to respective files.
28. Write a program to compare two given strings.

(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:

A screenshot of a cell phone

Description automatically generated

## (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:

# 

## (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:

A screenshot of a cell phone

Description automatically generated

## (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:

A close up of a logo

Description automatically generated

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);

}A screenshot of a cell phone

Description automatically generated

## Output:

## 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:

A screenshot of a cell phone

Description automatically generated

## (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:

A screenshot of a cell phone

Description automatically generated

## (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:

A close up of a logo

Description automatically generated

## (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:

A screenshot of a cell phone

Description automatically generated

## (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:

A screenshot of a cell phone

Description automatically generated

## (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 find 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);

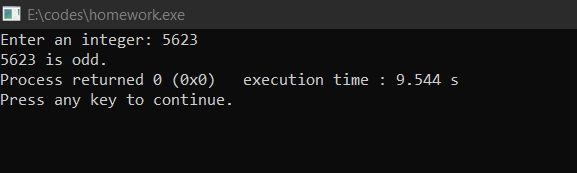
else

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

return 0;

}

## Output:



## 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);

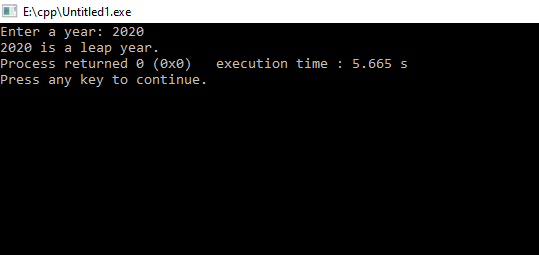
else

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

return 0;

}

## Output:



## (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");

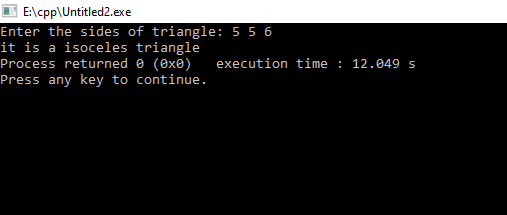
else

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

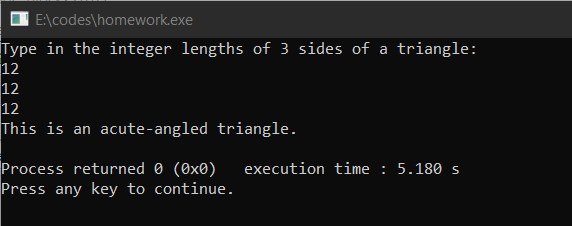
printf("Not a triangle\n");

}

return 0;

}

## Output:



## 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");

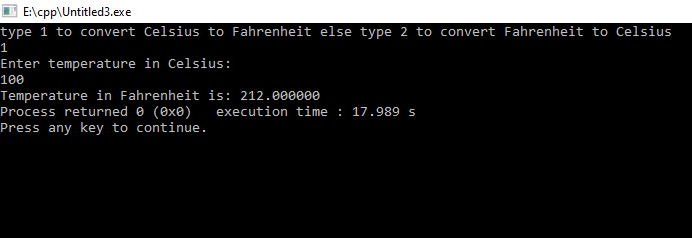
scanf("%f", &f);

c= (f-32) / 1.8;

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

return 0;}

## Output:



## (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);

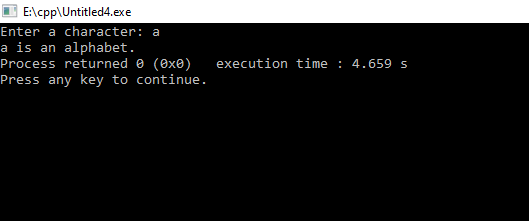
else if(c>='1'&& c<='9')

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'))

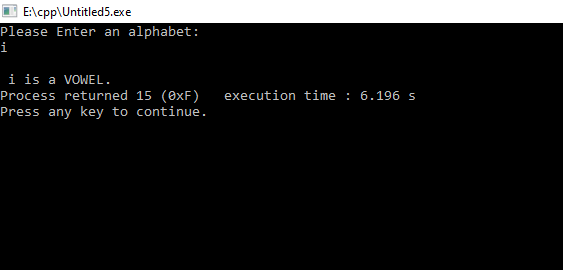
printf("\n %c is a VOWEL.", c);

else

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

}

## Output:



## (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

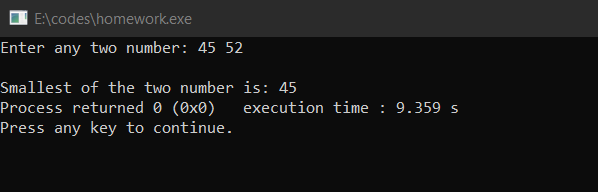
small=b;

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);

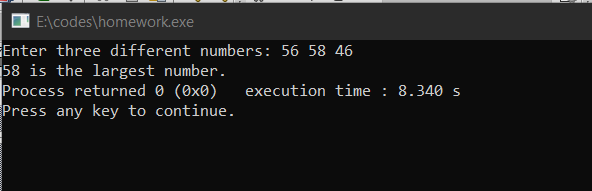
if (n3 >= n1 && n3 >= n2)

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

return 0;

}

## Output:



## 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.

## 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:

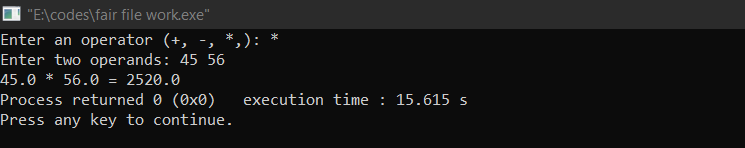
printf("Error! operator is not correct");

}

return 0;

}

## Output:



## 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.

## 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);

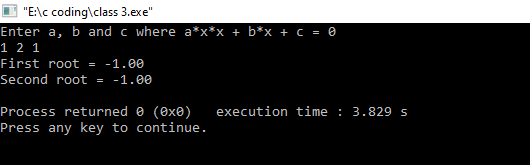
printf("Second root = %.2lf\n", root2);

}

return 0;

}

## Output:



## 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.

## 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)

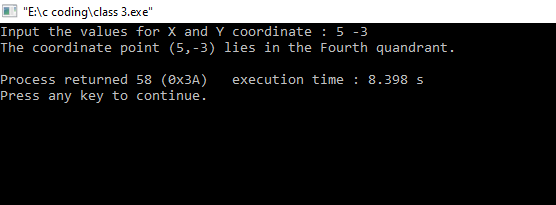
printf("The coordinate point (%d,%d) lies in the Fourth quandrant.\n",co1,co2);

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

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

}

## Output:



## 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;

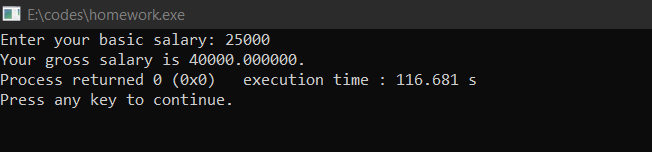
gross\_salary = basic\_salary+da+hra;

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

return 0;

}

## Output:



## 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);

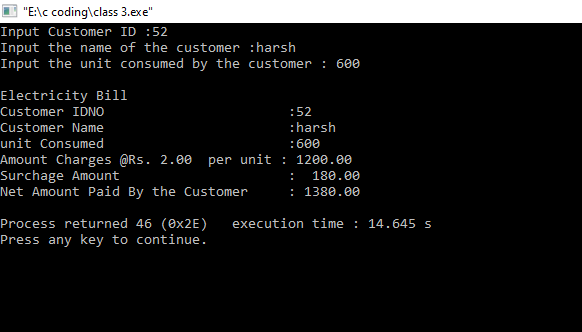
printf("unit Consumed :%d\n",conu);

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:



## 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 if else 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;

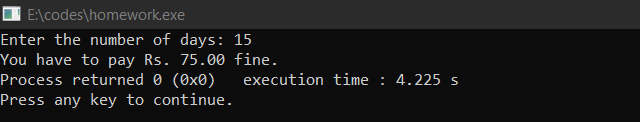
if (days > 30)

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

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

return0;}

## Output:



## 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.

## 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;

}

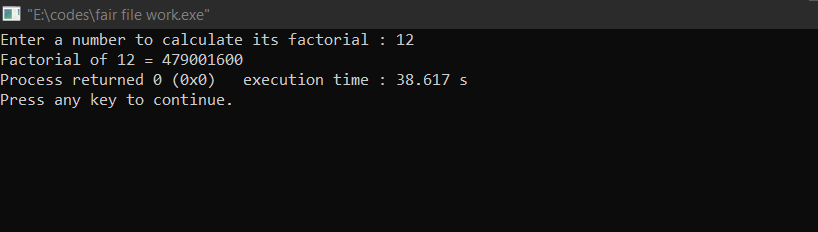
printf("Factorial of %d = %d", num, fact);

}

return 0;

}

## Output:



## 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.

## 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:

## 

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

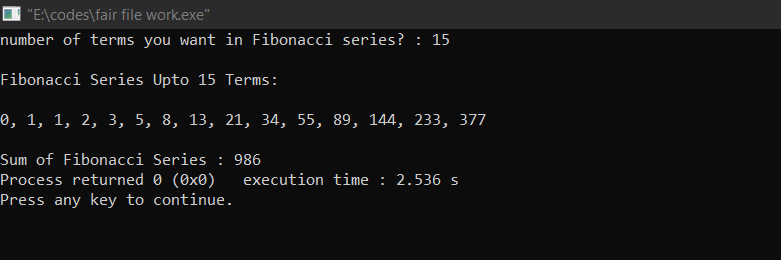
### Language used : C

### 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.

## 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:



## **21.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;

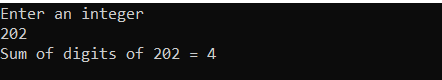
}

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

return 0;

}

OUTPUT:-



## **22.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: C Programming Operators, C if...else Statement, C while Loop is used.

## 

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)

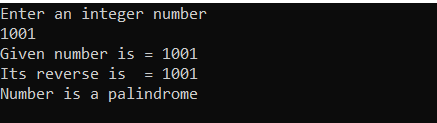
printf("Number is a palindrome \n");

else

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

}

OUTPUT:



## **23.Write a program in C to accept an integer numbers and to check a number 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

## **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:



## **24.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.

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



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

Language used- C

Theory- the concept of loops will be used here. Everytime when the loop executes the value of k increases by 2 and the total sum is stored in sum variable

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;

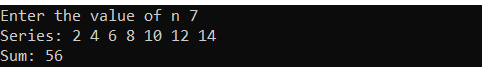
}

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

return 0;

}

OUTPUT:



## **26.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:



## **27.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 loops 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:



## **28.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:



## **29.Write a program to find the sum of following series:**

S = -13+ 33-53+ 73–93+ 113-........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:



## **30.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



## **31.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:



## **32.Write a program to find the sum of following series:**

## **S = 14 + 34 + 54 + 74 + …………….. 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:

Series: 1,3,5,7,9,11,13,15,17,19,21,23,25,27,29,31,33,35,37,39,41,43,45,47,49,51,53,55,57,59,61,63,65,67,69,71,73,75,77,79,81,83,85,87,89,91,93,95,97,99,101,103,105,107,109,111,113,115,117,119,121,123,125,127,129,131,133,135,137,139,141,143,145,147,149,151,153,155,157,159,161,163,165,167,169,171,173,175,177,179,181,183,185,187,189,191,193,195,197,199

Sum of 4th power of series: 1932562308

## **33.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");

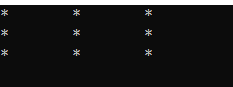
}

printf("\n");

}

}

OUTPUT:



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

## **1 2 3**

## **1 2 3**

## **1 2 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);

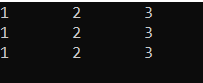
}

printf("\n");

}

}

OUTPUT:



## **35.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);

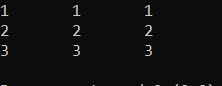
}

printf("\n");

}

}

OUTPUT:



## **36.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);

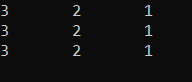
}

printf("\n");

}

}

OUTPUT:



## **37.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);

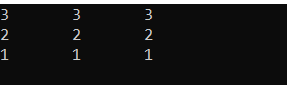
}

printf("\n");

}

}

OUTPUT:



## **38.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");

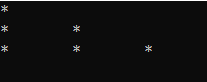
}

printf("\n");

}

}

OUTPUT:



## **39.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);

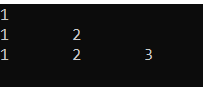
}

printf("\n");

}

}

OUTPUT:



## **40.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);

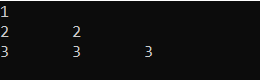
}

printf("\n");

}

}

OUTPUT:



## **41.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);

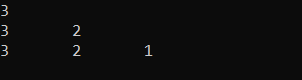
}

printf("\n");

}

}

OUTPUT:



## **42.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);

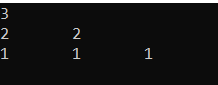
}

printf("\n");

}

}

OUTPUT:



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

## **\***

## **\* \* \***

## **\* \* \* \* \***

## **\* \* \* \* \* \* \***

CODE:

#include <stdio.h>

int main()

{

int i, space, rows, k = 0;

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

scanf("%d", &rows);

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

{

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

{

printf(" ");

}

while (k != 2 \* i - 1)

{

printf("\* ");

++k;

}

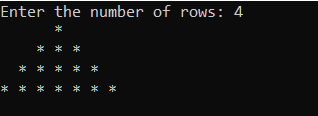
printf("\n");

}

return 0;

}

OUTPUT:



**44.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**

CODE:

#include<stdio.h>

int main()

{

int i,j,n,k;

printf("Enter a value : ");

scanf("%d",&n);

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");

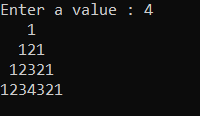
}

}

return 0 ;

}

OUTPUT:



## **45.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**

## CODE:

## **46.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**

CODE:

#include<stdio.h>

int main()

{

int i, j, rows ;

int count = 1;

printf("Enter the number of rows\n");

scanf("%d", &rows);

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

{

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

{

printf("%d ", count);

count = !count;

}

count = i % 2;

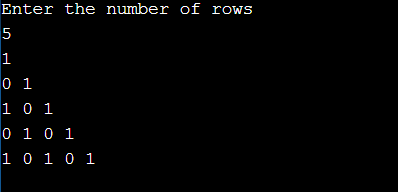
printf("\n");

}

return(0);

}

OUTPUT:



## **47.Write a program to print all prime numbers <= a given 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)

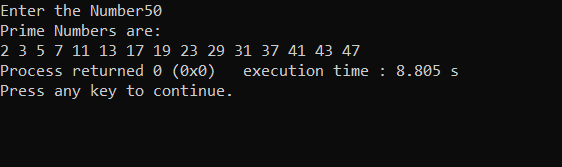
printf("%d " ,i);

}

return 0;

}

OUTPUT:



## **48.Write a program to convert Decimal no to Binary No.**

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--)

{

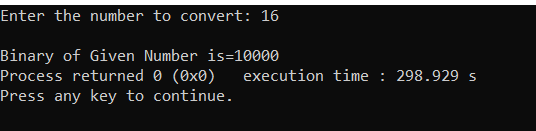
printf("%d",a[i]);

}

return 0;

}

OUTPUT



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

**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++)

{

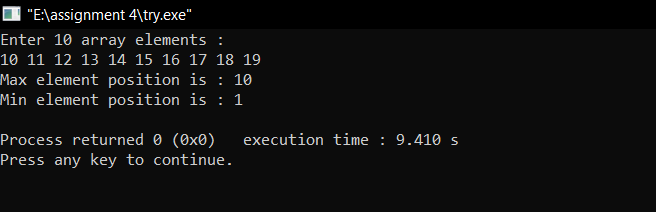
if(a[i]<a[temp2]) temp2=i;

}

printf("Min element position is : %d\n",temp2); return 0;

}

OUTPUT:-



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

**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;

}

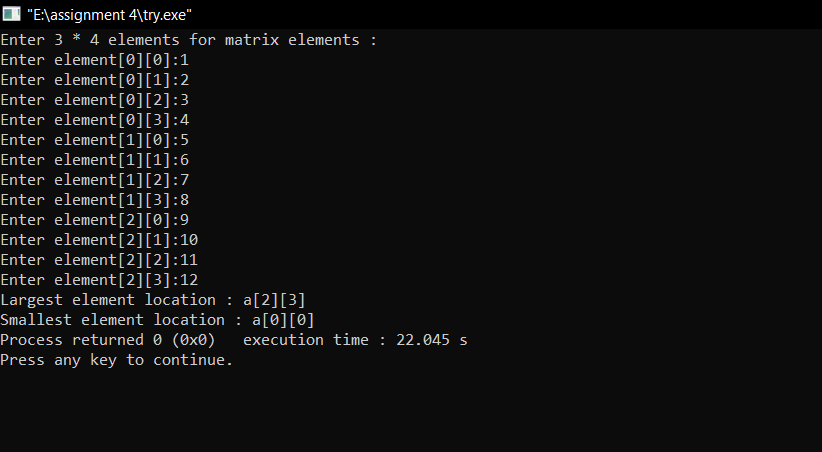
}

}

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 .

**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)

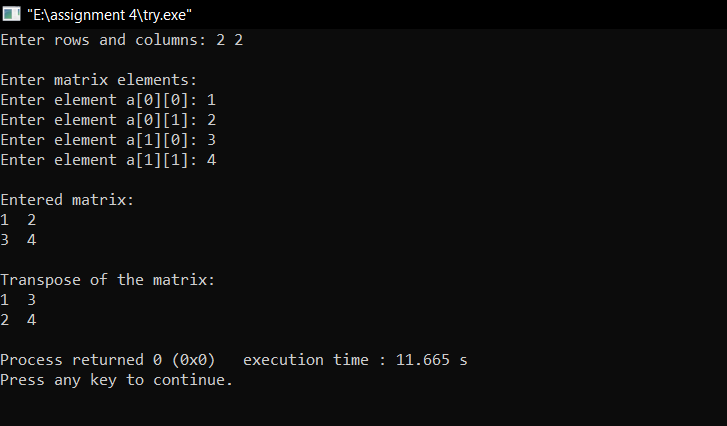
printf("\n");

}

return 0;

}

OUTPUT:-



## 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.

**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)

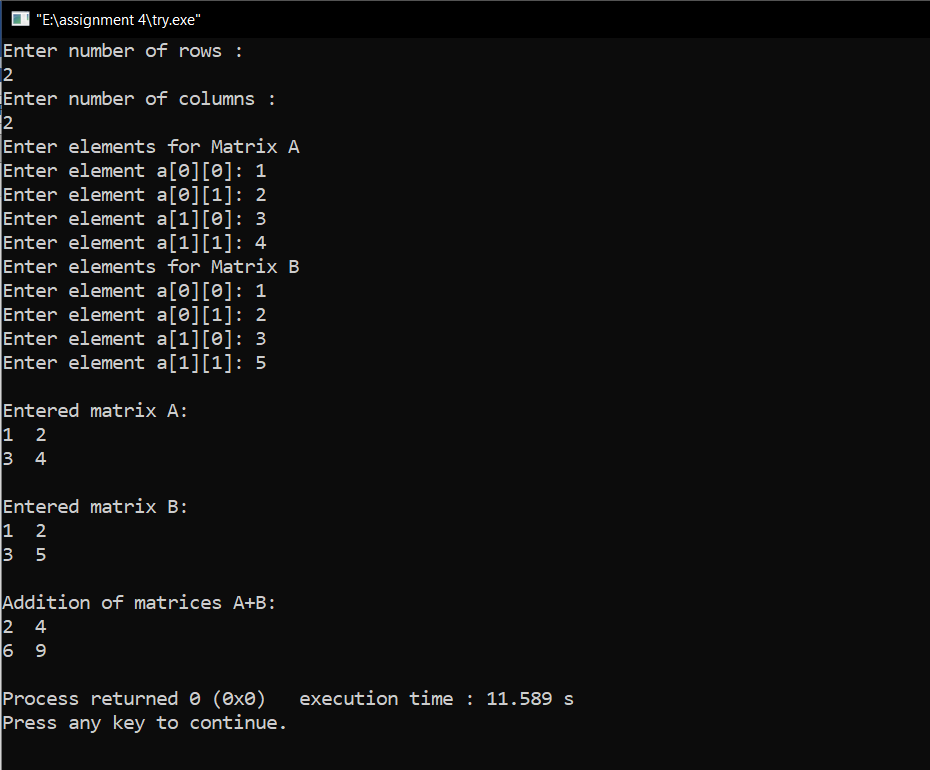
printf("\n");

}

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)

**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)

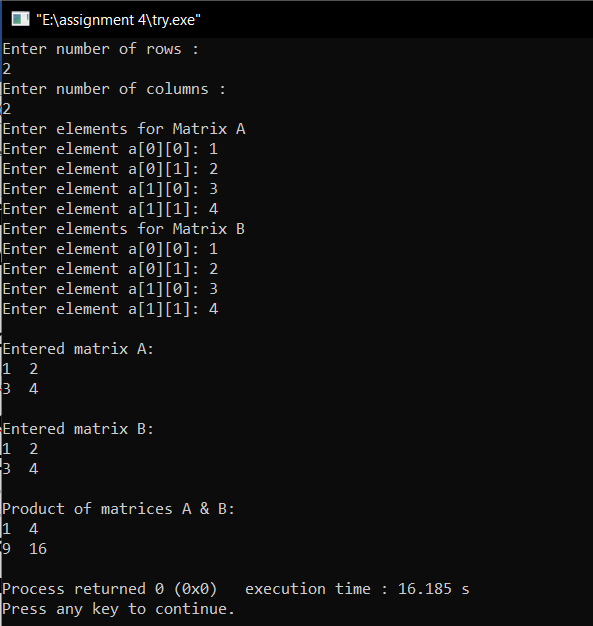
printf("\n");

}

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.

**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)

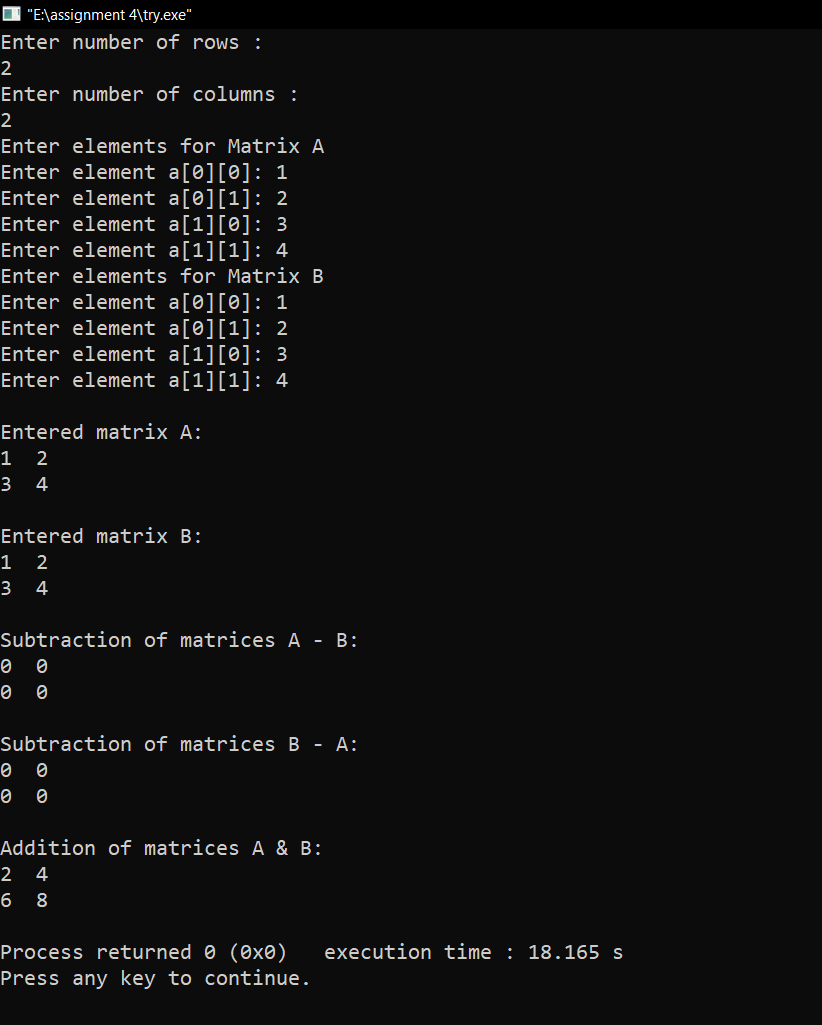
printf("\n");

}

return 0;

}

**P.T.O**

OUTPUT:-

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

**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);

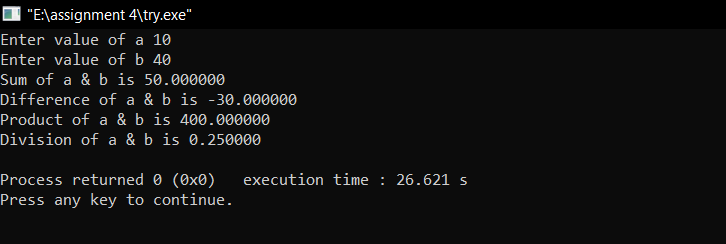
printf("Difference of a & b is %f\n",c); c=prod(a,b);

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

**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()

{

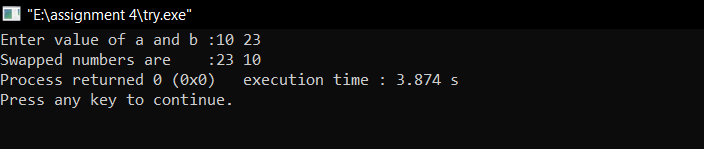
int a ,b;

printf("Enter value of a and b :"); scanf("%d %d",&a, &b); swap(a,b);

return 0;

}

## OUTPUT:



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

**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()

{

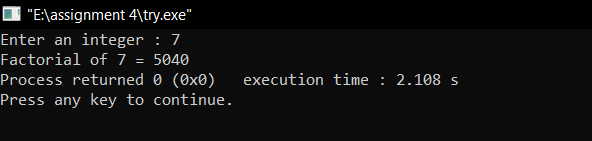
int n;

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

**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)

{

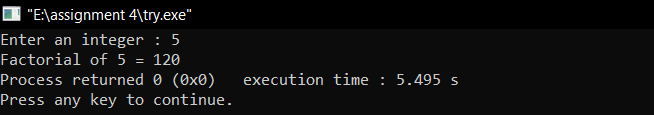
if(n>=1)

return n\*fact(n-1); else

return 1;

}

## OUTPUT:



## 59-Write a C program to check whether a number is even or odd using functions.

**Theory: A formal definition of an even number is that it is an integer of the form n = 2k, where k is an integer, it can then be shown that an odd number is an integer of the form n**

**= 2k + 1.**

## CODE:

#include<stdio.h> int eveodd(int n); int main()

{

int n;

printf("Enter an integer : "); scanf("%d",&n);

int x = eveodd(n); if(x==0)

{

printf("Number is even");

}

else

{

printf("Number is odd");

}

return 0;

}

int eveodd(int n)

{

if(n%2==0)

{

return 0;

}

else

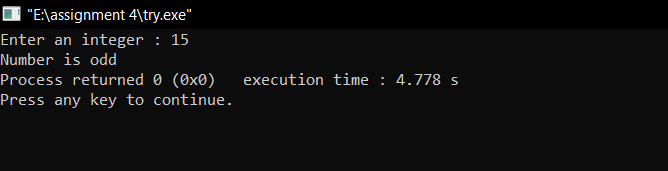
{

return 1;

}

}

## OUTPUT:



## 60-Write a C program to check whether a number is Prime, Armstrong or perfect number using functions.

**Theory: it is a program to check whether a number is Prime, Armstrong or perfect number using functions.**

## CODE:

#include<stdio.h> int isprime(int n);

int isArmstrong(int n); int isperfect(int n);

int main()

{

int n;

printf("Enter an integer : "); scanf("%d",&n); if(isprime(n)==1)

{

printf("The number is PRIME\n");

}

else

{

printf("The number is not PRIME\n");

}

if(isArmstrong(n) == n)

{

printf("The number is ARMSTRONG\n");

}

else

{

printf("The number is not ARMSTRONG\n");

}

if(isperfect(n) == n)

{

printf("The number is PERFECT\n");

}

else

{

printf("The number is not PERFECT\n");

}

return 0;

}

int isprime(int n)

{

if(n<=1) return 0;

for(int i=2;i<n;i++)

{

if(n%i == 0)

{

return 0;

}

}

return 1;

}

int isArmstrong(int n)

{

int original = n,result=0,remainder; while(original != 0){

remainder = original % 10;

result+= remainder \* remainder \* remainder; original /= 10;

}

return result;

}

int isperfect(int n)

{

int rem,sum=0,i; for(i=1;i<n;i++)

{

rem = n%i; if(rem==0)

{

sum = sum+i;

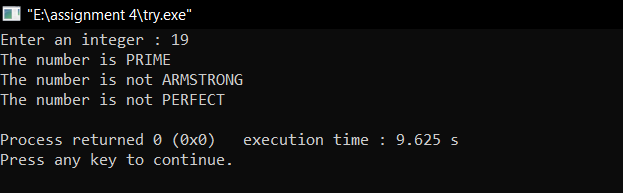
}

}

return sum;

}

## OUTPUT:



## 61-Write a C program to find all prime numbers between given interval using functions.

**Theory: In the below program, the range of numbers is taken as input and stored in the variables ‘lowerLimit’ and ‘upperLimit’. Then using while-loop, the numbers between the interval of lowerLimit and upperLimit are traversed. For each number in the for loop, it is checked if this number is prime or not. If found prime, print the number. Then the next number in the loop is checked, till all numbers are checked.**

## CODE:

#include <stdio.h> int isPrime(int num);

void printPrimes(int lowerLimit, int upperLimit); int main()

{

int lowerLimit, upperLimit;

printf("Enter the lower and upper limit to list primes: "); scanf("%d%d", &lowerLimit, &upperLimit); printPrimes(lowerLimit, upperLimit);

return 0;

}

void printPrimes(int lowerLimit, int upperLimit)

{

printf("All prime number between %d to %d are: ", lowerLimit, upperLimit);

while(lowerLimit <= upperLimit)

{

if(isPrime(lowerLimit))

{

printf("%d, ", lowerLimit);

}

lowerLimit++;

}

}

int isPrime(int num)

{

int i;

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

{

if(num % i == 0)

{

return 0;

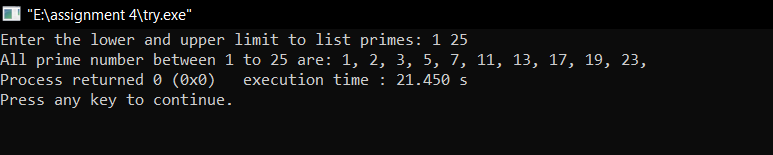
}

}

return 1;

}

## OUTPUT:



## 62-Write a C program to print all strong numbers between given interval using functions.

**Theory: *Strong number* is a special number whose sum of factorial of digits is equal to the original number.**

**For example: 145 is strong number. Since, 1! + 4! + 5! = 145**

## CODE:

#include <stdio.h> long long fact(int num);

void printStrongNumbers(int start, int end); int main()

{

int start, end;

printf("Enter the lower limit to find strong number: "); scanf("%d", &start);

printf("Enter the upper limit to find strong number: "); scanf("%d", &end);

printf("All strong numbers between %d to %d are: \n", start, end);

printStrongNumbers(start, end); return 0;

}

void printStrongNumbers(int start, int end)

{

long long sum; int num;

while(start != end)

{

sum = 0; num = start;

while(num != 0)

{

sum += fact(num % 10);

num /= 10;

}

if(start == sum)

{

printf("%d, ", start);

}

start++;

}

}

long long fact(int num)

{

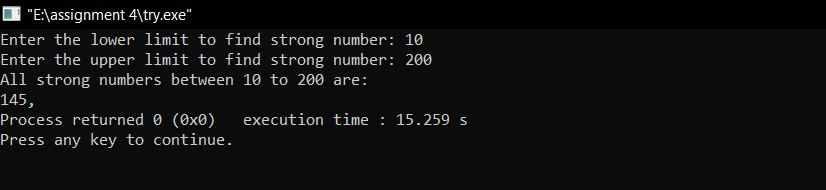
if(num == 0)

return 1; else

return (num \* fact(num-1));

}

## OUTPUT:



## 63-Write a C program to find power of any number using recursion

**Theory: We include one base case i.e. when exponent is zero then we return 1 and a non base case i.e. multiply base with recursive call to power with expopnent decreased by 1.**

## CODE:

#include <stdio.h>

double pow(double base, int expo); int main()

{

double base, power; int expo; printf("Enter base: "); scanf("%lf", &base);

printf("Enter exponent: "); scanf("%d", &expo); power = pow(base, expo);

printf("%.2lf ^ %d = %f", base, expo, power); return 0;

}

double pow(double base, int expo)

{

if(expo == 0)

return 1;

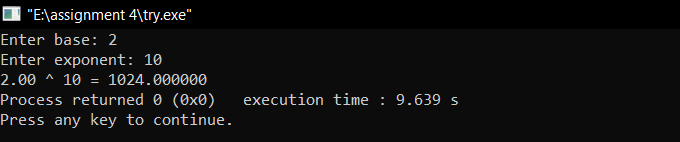
else if(expo > 0)

return base \* pow(base, expo - 1); else

return 1 / pow(base, -expo);

}

## OUTPUT:



## 64-Declare a structure name student containing members name, roll\_no, marks. Create an array of 30 such students. Write a program to read and display the contents of array.

**Theory: In this program, a structure student is created. The structure has three members: name (string), roll no. (integer) and marks (int).**

**Then, a structure variable a of size 30 is created to store information and display it on the screen.**

## CODE:

#include<stdio.h> struct student{

char name[20]; int roll\_no;

int marks;

};

int main()

{

struct student a[30]; for(int i=0;i<30;i++)

{

printf("\nEnter name of student : "); scanf("%s",a[i].name); printf("\nEnter roll\_no of student : "); scanf("%d",&a[i].roll\_no); printf("\nEnter marks of student : "); scanf("%d",&a[i].marks);

}

for(int i=0;i<30;i++)

{

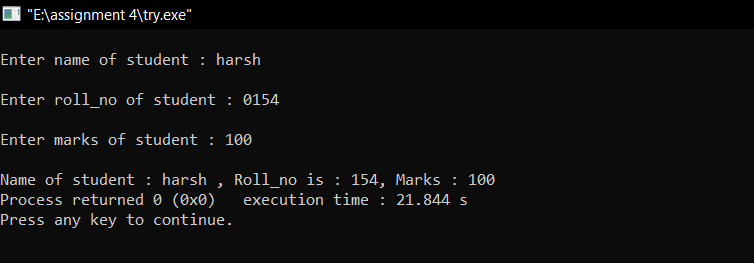
printf("\nName of student : %s , Roll\_no is : %d, Marks :

%d",a[i].name,a[i].roll\_no,a[i].marks);

}

}

## OUTPUT:



(for only 1 entry)\*

## 65-Write a simple database program in C which stores personal details of 100 persons such as Name, Date of Birth, Address, Phone number etc.

**Theory: In this program, a structure person is created. The structure has four members: name (string), DOB (integer dd, mm, yyyy), address (string) and phone number (long long).**

**Then, a structure variable a of size 100 is created to store information and display it on the screen.**

## CODE:

#include<stdio.h> struct person{

char name[20]; int dd,mm,yyyy;

char address[100]; long long phone;

};

int main()

{

struct person a[100]; for(int i=0;i<100;i++)

{

printf("\nEnter name of person : "); scanf("%s",a[i].name);

printf("\nEnter dob of person giving spaces format like dd mm yyyy: ");

scanf("%d",&a[i].dd);

scanf("%d",&a[i].mm);

scanf("%d",&a[i].yyyy); printf("\nEnter address of person : "); scanf("%s",a[i].address);

printf("\nEnter phone number of person : "); scanf("%lld",&a[i].phone);

}

for(int i=0;i<100;i++)

{

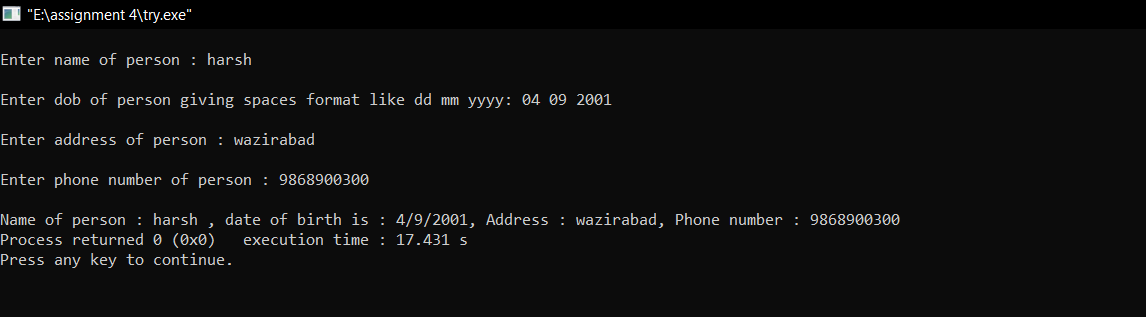
printf("\nName of person : %s , date of birth is : %d/%d/%d, Address : %s, Phone number :

%lld",a[i].name,a[i].dd,a[i].mm,a[i].yyyy,a[i].address,a[i].phon e);

}

}

## OUTPUT:



(for only 1 entry)\*

## 66-Write a program in ‘C’ that compares two given dates. To store a date, use a struct that contains three members namely day, month, and year. If the dates are equal, then display message as “equal” otherwise “Unequal”.

**Theory: To store date use structure say date that contains three members namely date, month and year. If the dates are equal then display message as "Equal" otherwise "Unequal".**

## CODE:

#include<stdio.h> #include<string.h> struct date

{

int date; int month; int year;

};

int main ()

{ int i,f=0; struct date d[2]; for(i=0;i<2;i++)

{ printf("Enter day %d :”,i+1); scanf("%d",&d[i].date); printf("Enter the month %d :",i+1); scanf("%d",&d[i].month); printf("Enter the year %d :",i+1); scanf("%d",&d[i].year);

}

if(d[0].date==d[1].date)

{ if(d[0].month==d[1].month)

{ if(d[0].year==d[1].year)

{ f=1;

}

}

}

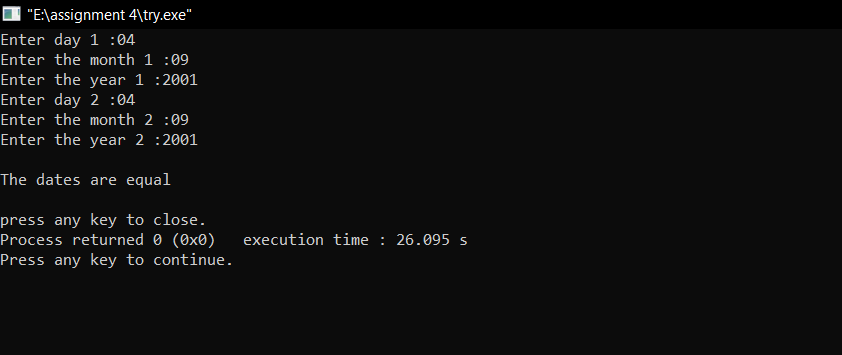
if(f==1)

printf("\nThe dates are equal"); else

printf("\nThe dates are not equal"); printf ("\n\npress any key to close.");

}

## OUTPUT:



## 67-Write a program which reads your name from the keyboard and outputs a list of ASCII codes, which represent your name.

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

## CODE:

#include<stdio.h> int main()

{ int i;

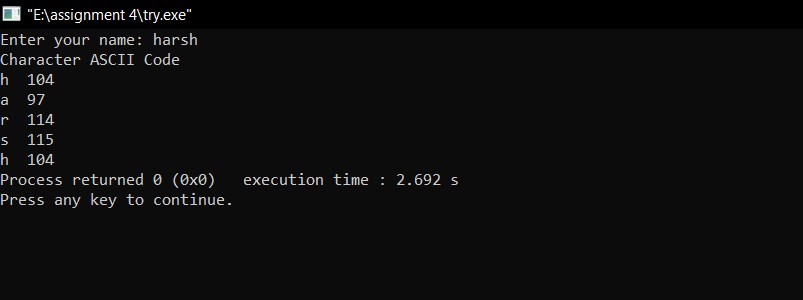
char name[50]; printf("Enter your name: "); gets(name);

printf("Character ASCII Code"); for(i=0;name[i]!=NULL;i++)

printf("\n%c %d",name[i],name[i]); return 0;

}

## OUTPUT:



## 68-Write a program which will read a text and count all occurrences of all characters which are part of text.

**Theory: The concept of string, it is nothing but an array of characters. The value of a string is determined by the terminating character. Its value is considered to be 0.**

## CODE:

#include <stdio.h> #include <string.h> int main()

{

char s[1000];

int i,j,k,count=0,n; printf("Enter the string : "); gets(s);

for(j=0;s[j];j++); n=j;

printf(" frequency count character in string:\n"); for(i=0;i<n;i++)

{ count=1; if(s[i])

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

{ if(s[i]==s[j])

{ count++; s[j]='\0';

}

}

printf(" '%c' = %d \n",s[i],count);

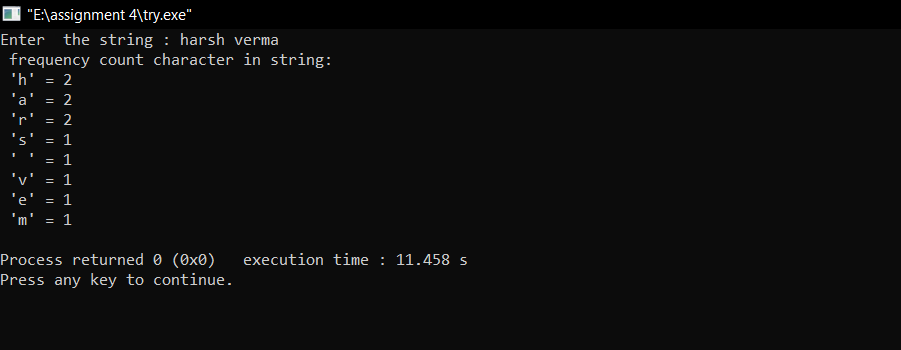
}

}

return 0;

}

## OUTPUT:



## 69-Write a program which will read a text and count all occurrences of a particular word.

**Theory: Count=0,Read the entered string as gets(s). Read the entered character c as getchar().**

**Compare the entered character with the elements of the string using for loop with the structure for(i=0;s[i];i++).**

**If the character match with s[i] then increase the count value by 1.**

**For each occurrence of the character, the count value will be increased. So after all iterations of for loop, we will get total occurrences of the character in the string.**

## CODE:

#include <stdio.h> int main()

{ char s[1000],c; int i,count=0;

printf("Enter the string : "); gets(s);

printf("Enter character to be searched: "); c=getchar();

for(i=0;s[i];i++)

{

if(s[i]==c)

{

count++;

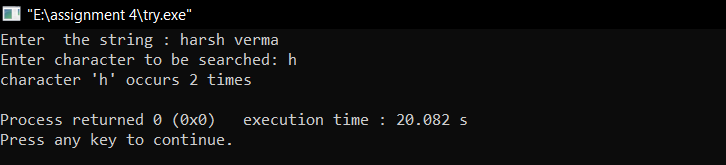
}

}

printf("character '%c' occurs %d times \n ",c,count); return 0;

}

## OUTPUT:



## 70- Write a program which reads a string from the keyboard and determines whether the string is a palindrome (Ignore Capitalization)

**Theory: A string is a palindrome if it reads same from forward as well as backward. This is a C program to check whether a string is palindrome or not.**

## CODE:

#include <stdio.h> #include <string.h> int main()

{

char text[20], reverse\_text[20]; int i,n, length = 0; printf("Enter text: "); gets(text);

for (i = 0; text[i] != '\0'; i++)

{

length++;

}

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

{

reverse\_text[length - i - 1] = text[i];

}

for (n = 1, i = 0; i < length; i++)

{

if (reverse\_text[i] != text[i]) n = 0;

}

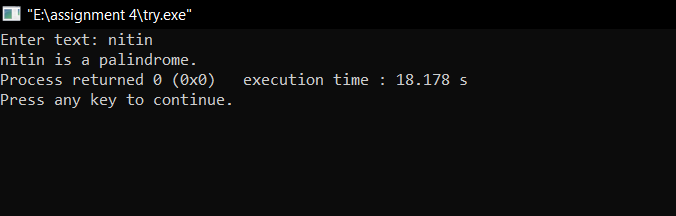
if (n == 1)

printf("%s is a palindrome.", text); else

printf("%s is not a palindrome", text); return 0;

}

## OUTPUT:



## 71-Write macro definition with arguments for calculation of simple interest and amount. Store these macro definitions in a file called ‘Interest.h”. Include this file in your program and use the macro definition for calculating simple interest and amount.

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

## CODE:

#define SI(p, t, r) ( (p \* t \* r) / 100.0 ) #define AMT(p, t, r) ( SI(p, t, r) + p )

#include<stdio.h> #include "interest.h" int main()

{

float p, t, r;

printf("Enter principal amount\n"); scanf("%f", &p);

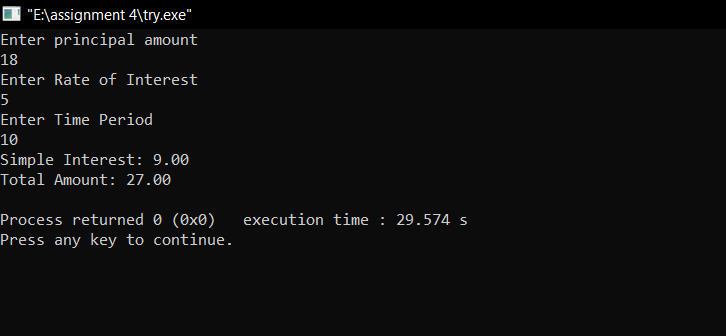
printf("Enter Rate of Interest\n"); scanf("%f", &r);

printf("Enter Time Period\n"); scanf("%f", &t);

printf("Simple Interest: %0.2f\n", SI(p, t, r)); printf("Total Amount: %0.2f\n", AMT(p, t, r)); return 0;

}

## OUTPUT:



## 72-Write a program to copy the contents of one file to another file.

**Theory: Copying the content of source file to target file.**

## CODE:

#include <stdio.h> #include <stdlib.h> int main()

{

char ch, source\_file[20], target\_file[20]; FILE \*source, \*target;

printf("Enter name of file to copy\n"); gets(source\_file);

source = fopen(source\_file, "r"); if( source == NULL )

{

printf("Press any key to exit...\n"); exit(EXIT\_FAILURE);

}

printf("Enter name of target file\n"); gets(target\_file);

target = fopen(target\_file, "w"); if( target == NULL )

{

fclose(source);

printf("Press any key to exit...\n"); exit(EXIT\_FAILURE);

}

while( ( ch = fgetc(source) ) != EOF ) fputc(ch, target);

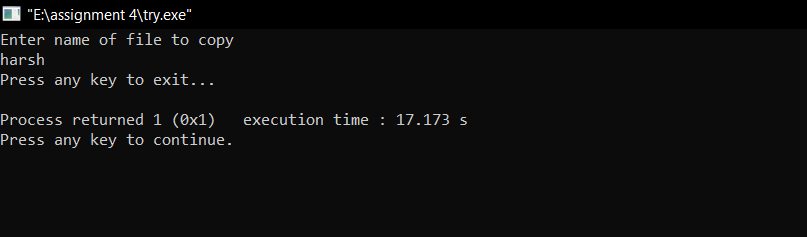
printf("File copied successfully.\n"); fclose(source);

fclose(target);

return 0;

}

## OUTPUT:



## 73-Write a program which will store ten integers to one file and squares of these numbers to another file.

**Theory: Copying the content of source file to target file by squaring the numbers.**

## CODE:

#include <stdio.h> #include <stdlib.h> int main()

{

int i, ch, source\_file[10], target\_file[10]; FILE \*source, \*target;

printf("Enter any 10 numbers\n"); for(i = 0; i<10;i++)

{ printf("Enter no. %d : \n",i+1); scanf("%d",&source\_file[i]);

}

source = fopen(source\_file, "r"); if( source == NULL )

{

printf("Press any key to exit...\n"); exit(EXIT\_FAILURE);

}

printf("Enter name of target file\n"); gets(target\_file);

target = fopen(target\_file, "w"); if( target == NULL )

{

fclose(source);

printf("Press any key to exit...\n"); exit(EXIT\_FAILURE);

}

while( ( ch = fgetc(source) ) != EOF )

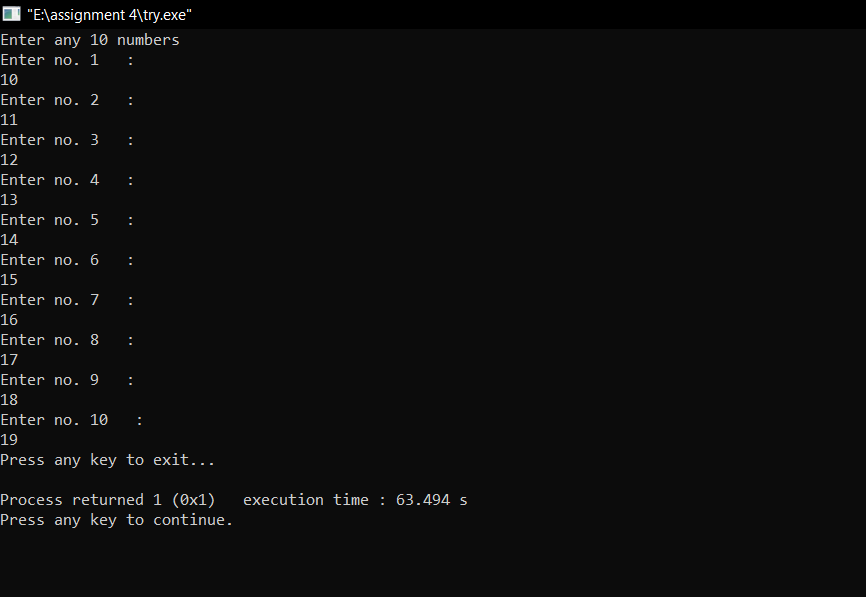
fputc(ch\*ch, target);

printf("File copied successfully.\n"); fclose(source);

fclose(target); return 0;

}

## OUTPUT:



## 74-Write a program which will store ten integers to one file and stores the odd and even numbers to respective files

**Theory: Copying the content of source file to ODD and EVEN target files.**

## CODE:

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

int main()

{ int a,n,i;

FILE \*fp1,\*fp2,\*fp3; fp1=fopen("DATA","w"); if(fp1==NULL)

{

printf("File could not open!!"); exit(0);

}

printf("How many numbers?"); scanf("%d",&n);

printf("Enter contents of DATA file: "); for(i=0;i<n;++i)

{

scanf("%d",&a); putw(a,fp1);

}

fclose(fp1); fp1=fopen("DATA","r");

fp2=fopen("ODD","w");

fp3=fopen("EVEN","w"); if(fp1==NULL||fp2==NULL||fp3==NULL)

{

printf("File could not open!!"); exit(0);

}

while((a=getw(fp1))!=EOF)

{

if(a%2!=0) putw(a,fp2);

else

putw(a,fp3);

}

fclose(fp1); fclose(fp2); fclose(fp3);

fp2=fopen("ODD","r");

fp3=fopen("EVEN","r"); if(fp2==NULL||fp3==NULL)

{ printf("File could not open!!"); exit(0);

}

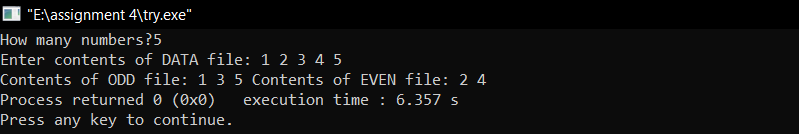
printf("Contents of ODD file: "); while((a=getw(fp2))!=EOF)

printf("%d ",a); printf("Contents of EVEN file: "); while((a=getw(fp3))!=EOF)

printf("%d ",a); fclose(fp2); fclose(fp3); return 0;

}

## OUTPUT:



## 75-Write a program to compare two given strings.

**Theory: The string function which is pre-defined in a string.h header file is**

**a strcmp() function. The strcmp() function consider two strings as a parameter, and this function returns an integer value where the integer value can be zero, positive or negative.**

## CODE:

#include <stdio.h> #include<string.h> int main()

{

char str1[20]; char str2[20]; int value;

printf("Enter the first string : "); scanf("%s",str1);

printf("Enter the second string : "); scanf("%s",str2); value=strcmp(str1,str2); if(value==0)

printf("strings are same"); else

printf("strings are not same"); return 0;

}

## OUTPUT:

