| S.No: 1 | Exp. Name: ***Program to print simple statement.*** | Date: 2023-09-14 |
|---|---|---|

**Aim:**
Write a C program to print a simple message.

**ALGORITHM:**
Step 1: Start
Step 2: Display "Welcome to C Programming".
Step 3: Stop.

**Source Code:**

Hello.c

```
#include<stdio.h>
main()
{
        printf("Welcome to C programming");
}
```

## Execution Results - All test cases have succeeded!

| Test Case - 1 |
|---|
| **User Output** |
| Welcome to C programming |


ID: S1331
2015-2030-Faculty-CSE
Kakatiya Institute of Technology and Science

| S.No: 2 | Exp. Name: ***Program to display all the primitive data type variables.*** | Date: 2023-09-14 |
|---|---|---|

**Aim:**

Write a C Programdemonstratingto display all the primitive data type variables.

**ALGORITHM:**

Step 1:Start
Step2:input dec=5, ch='s', pi=3.14
Step 3:print dec, ch, pi
Step 4:stop Program

**Source Code:**

program.c

```
#include<stdio.h>
int main()
{
        int g,dec=5;
        char ch='p';
        float pi=3.14;
        g=2*dec;
        printf("%d %f %c",g,pi,ch);
}
```

## Execution Results - All test cases have succeeded!

| Test Case - 1 |
|---|
| **User Output** |
| 10 3.140000 p |

| S.No: 3 | Exp. Name: ***Program for demonstrating reading different values using scanf() and display them*** | Date: 2023-09-14 |
|---|---|---|

**Aim:**

Write a program for demonstrating reading different values usingscanf() and display them

**Source Code:**

format5.c

```
#include<stdio.h>
void main()
{
        char a;
        int b;
        float c;
        printf("Enter a character: ");
        scanf("%c",&a);
        printf("Enter the integer: ");
        scanf("%d",&b);
        printf("Enter the float value: ");
        scanf("%f",&c);
        printf("Character: %c\n",a);
        printf("Integer value: %d\n",b);
        printf("Float value: %f\n",c);
}
```

## Execution Results - All test cases have succeeded!

| Test Case - 1 |
|---|
| **User Output** |
| Enter a character: |
| A |
| Enter the integer: |
| 54 |
| Enter the float value: |
| 54.654 |
| Character: A |
| Integer value: 54 |
| Float value: 54.653999 |

| S.No: 4 | Exp. Name: ***Program to calculate simple interest*** | Date: 2023-09-14 |
|---|---|---|

**Aim:**
Write a C Program to calculate simple interest.

**ALGORITHM:**
Step 1: Start
Step 2: Inputp, t, r
Step 3: Calculate Six(p*t*r)/100
Step 4: Print simple interest SI
Step 5: Stop

**Source Code:**

si.c

```
#include <stdio.h>
void main() {
        float p, r, t, si;
        printf("Enter principal,rate and time: ");
        scanf("%f %f %f", &p, &r, &t);
        si = (p * r * t) / 100;
        printf("Simple Interest = %f\n", si);
}
```

## Execution Results - All test cases have succeeded!

| Test Case - 1 |
|---|
| **User Output** |
| Enter principal,rate and time: |
| 30000 3 4 |
| Simple Interest = 3600.000000 |

| Test Case - 2 |
|---|
| **User Output** |
| Enter principal,rate and time: |
| 5521036 10 5 |
| Simple Interest = 2760518.000000 |

| S.No: 5 | Exp. Name: ***Program to read length and breadth of a rectangle as input and find its area and perimeter.*** | Date: 2023-09-14 |
|---|---|---|



**Aim:**

Write a C Program to read the length and breadth of a rectangle as input and find its area and perimeter.

**ALGORITHM:**

Step 1: Start
Slept:2 Input length and breadth
Step 3: Calculate area ‹- length * breadth
Step 4:Calculateperimeter ‹-2.0*(length + breadth)
Step 5: Print area and perimeter
Step 6: Stop

**Source Code:**

length.c

```
#include <stdio.h>
main()
{
        float length,width,area,perimeter;
        printf("Enter the length & width of the rectangle: ");
        scanf("%f%f",&length,&width);
        area = length*width;
        perimeter= 2.0*(length+width);
        printf("The Perimeter and area of rectangle = %f, %f units\n", perimeter,area);
}
```

## Execution Results - All test cases have succeeded!

| Test Case - 1 |
|---|
| **User Output** |
| Enter the length & width of the rectangle: |
| 5 8 |
| The Perimeter and area of rectangle = 26.000000, 40.000000 units |

| Test Case - 2 |
|---|
| **User Output** |
| Enter the length & width of the rectangle: |
| 28 10 |
| The Perimeter and area of rectangle = 76.000000, 280.000000 units |

| S.No: 6 | Exp. Name: ***Program to find the area and perimeter of the circle when the radius is given as input.*** | Date: 2023-09-14 |
|---|---|---|



**Aim:**

Write a C Program to find the area and perimeter of the circle when the radius is given as input.

**ALGORITHM:**

Step 1: Start
Step 2: Input radius (r), Define PI as a constant with a value of 3.14f
Step 3: Calculate areas PI * r * r
Step 4: Calculateperimeter2.0*PI * r
Step 5: Print area, perimeter
Step 6: Stop

**Source Code:**

radius.c

```
#include <stdio.h>
#define PI 3.14f
main()
{
        float radius, area, peri;
        printf("Enter radius of circle: ");
        scanf("%f",&radius);
        area = PI*radius*radius;
        peri = 2*(PI*radius);
        printf("Area of circle: %f\n", area);
        printf("Perimeter of circle: %f\n", peri);
}
```

## Execution Results - All test cases have succeeded!

| Test Case - 1 |
|---|
| **User Output** |
| Enter radius of circle: |
| 4.8 |
| Area of circle: 72.345612 |
| Perimeter of circle: 30.144003 |

| Test Case - 2 |
|---|
| **User Output** |
| Enter radius of circle: |
| 18.69 |
| Area of circle: 1096.852661 |
| Perimeter of circle: 117.373207 |

| S.No: 7 | Exp. Name: ***Program to display an integer value as left and right justified, a float value with required precision and a char with its equivalent ASCII value.*** | Date: 2023-09-14 |
|---|---|---|









**Aim:**

Write a C Program to display an integer value as left and right justified, a float value with requiredprecision and a char with its equivalent ASCII value.

**Source Code:**

ascii.c

```
#include <stdio.h>
void main() {
        int a, b;
        float c,d;
        char ch;
        printf("Enter two integers: ");
        scanf("%d %d", &a, &b);
        printf("Enter two float values: ");
        scanf("%f %f", &c, &d);
        printf("Enter a character: ");
        scanf(" %c", &ch);
        printf("Integer with left justified: %d\n", a);
        printf("Integer with right justified: %3d\n", b);
        printf("Float value with precision: %03f\n", c);
        printf("Float value with precision: %3.2f\n", d);
        printf("ASCII value of character: %d\n", ch);
}
```

## Execution Results - All test cases have succeeded!

| Test Case - 1 |
|---|
| **User Output** |
| Enter two integers: |
| 3 8 |
| Enter two float values: |
| 2.3 6.8 |
| Enter a character: |
| A |
| Integer with left justified: 3 |
| Integer with right justified:  8 |
| Float value with precision: 2.300000 |
| Float value with precision: 6.80 |
| ASCII value of character: 65 |

| Test Case - 2 |
|---|
| **User Output** |
| Enter two integers: |

| |
|---|
| 59 102 |
| Enter two float values: |
| 18.3647 57.21479 |
| Enter a character: |
| G |
| Integer with left justified: 59 |
| Integer with right justified: 102 |
| Float value with precision: 18.364700 |
| Float value with precision: 57.21 |
| ASCII value of character: 71 |

| S.No: 8 | Exp. Name: ***Program to display given integer in hexadecimal and octal formats*** | Date: 2023-09-14 |
|---|---|---|



**Aim:**

Write a C program to display the given integer in decimal, hexadecimal and octal formats.

**Source Code:**

formats.c

```
#include<stdio.h>
void main()
{
        int n1;
        scanf("%d",&n1);
        printf("Given integer value in Decimal form: %d\n",n1);
        printf("Given integer value in hexadecimal form: %X\n",n1);
        printf("Given integer value in Octal form: %o\n",n1);
}
```

## Execution Results - All test cases have succeeded!

| Test Case - 1 |
|---|
| **User Output** |
| 29 |
| Given integer value in Decimal form: 29 |
| Given integer value in hexadecimal form: 1D |
| Given integer value in Octal form: 35 |

| Test Case - 2 |
|---|
| **User Output** |
| 43 |
| Given integer value in Decimal form: 43 |
| Given integer value in hexadecimal form: 2B |
| Given integer value in Octal form: 53 |

| S.No: 9 | Exp. Name: ***Program to read two values and do the arithmetic operations.*** | Date: 2023-09-14 |
|---|---|---|





**Aim:**

Write a C program to read two values and do the arithmetic operations ( +, - , * , / , % )

**Source Code:**

1.c

```
#include <stdio.h>
void main() {
        int a, b, c;
        printf("Enter a value: ");
        scanf("%d", &a);
        printf("Enter b value: ");
        scanf("%d", &b);
        c = a + b;
        printf("Addition: %d\n", c);
        c = a - b;
        printf("Subtraction: %d\n", c);
        c = a * b;
        printf("Multiplication: %d\n", c);
        c = a / b;
        printf("Division: %d\n", c);
        c = a % b;
        printf("Modulus: %d\n", c);
        }
```

## Execution Results - All test cases have succeeded!

| Test Case - 1 |
|---|
| **User Output** |
| Enter a value: |
| 12 |
| Enter b value: |
| 3 |
| Addition: 15 |
| Subtraction: 9 |
| Multiplication: 36 |
| Division: 4 |
| Modulus: 0 |

| Test Case - 2 |
|---|
| **User Output** |
| Enter a value: |
| 18 |
| Enter b value: |
| 4 |

| Addition: 22 |
| --- |
| Subtraction: 14 |
| Multiplication: 72 |
| Division: 4 |
| Modulus: 2 |

| S.No: 10 | Exp. Name: ***Program to demonstrate the use of relational operators. ( <= , > , != )*** | Date: 2023-09-14 |
|---|---|---|









**Aim:**

Write a C program to demonstrate the use of relational operators. ( <= , > , != )

**Source Code:**

1.c

```
#include <stdio.h>
#include <conio.h>
void main() {
        int a, b;
        printf("Enter a value: ");
        scanf("%d", &a);
        printf("Enter b value: ");
        scanf("%d", &b);
        printf("A<=B: %d\n", (a<=b));
        printf("A>B: %d\n", (a>b));
        printf("A!=B: %d\n", (a!=b));
}
```

## Execution Results - All test cases have succeeded!

| Test Case - 1 |
|---|
| **User Output** |
| Enter a value: |
| 12 |
| Enter b value: |
| 3 |
| A<=B: 0 |
| A>B: 1 |
| A!=B: 1 |

| Test Case - 2 |
|---|
| **User Output** |
| Enter a value: |
| 324 |
| Enter b value: |
| 116 |
| A<=B: 0 |
| A>B: 1 |
| A!=B: 1 |

| S.No: 11 | Exp. Name: ***Program demonstrating conditional operator for finding the largest value between two given values*** | Date: 2023-09-14 |
|---|---|---|

**Aim:**

Write a C program which takes two integer values as input and displays the largest number.

**Source Code:**

largest.c

```
#include <stdio.h>
void main() {
        int a, b, big;
        scanf("%d %d", &a, &b);
        big = a > b ?  a : b;
        printf("largest = %d\n", big);
}
```

## Execution Results - All test cases have succeeded!

| Test Case - 1 |
|---|
| **User Output** |
| 5 |
| 8 |
| largest = 8 |

| Test Case - 2 |
|---|
| **User Output** |
| 14 |
| 8 |
| largest = 14 |

| S.No: 12 | Exp. Name: ***Program to output an integer in decimal format and a floating point number in float format.*** | Date: 2023-09-14 |
|---|---|---|









**Aim:**

Write a C program to output an integer in decimal format and a floating-point number in floating format. Experiment with outputting an integer in float format and vice versa.

**Source Code:**

conversion.c

```
#include <stdio.h>
#include <stdlib.h>
void main() {
        int decimal;
        float floating;
        printf("Enter an integer value: ");
        scanf("%d", &decimal);
        printf("Enter a floating point number: ");
        scanf("%f", &floating);
        printf("Integer value in decimal format = %d\n", decimal);
        printf("Integer value in floating format = %f\n", (float)decimal);
        printf("Floating value in decimal format = %d\n", (int)floating);
        printf("Floating value in floating Format = %f\n", floating);
}
```

## Execution Results - All test cases have succeeded!

| Test Case - 1 |
|---|
| **User Output** |
| Enter an integer value: |
| 34 |
| Enter a floating point number: |
| 8.98 |
| Integer value in decimal format = 34 |
| Integer value in floating format = 34.000000 |
| Floating value in decimal format = 8 |
| Floating value in floating Format = 8.980000 |

| Test Case - 2 |
|---|
| **User Output** |
| Enter an integer value: |
| 258 |
| Enter a floating point number: |
| 324.128 |
| Integer value in decimal format = 258 |
| Integer value in floating format = 258.000000 |
| Floating value in decimal format = 324 |
| Floating value in floating Format = 324.127991 |

| S.No: 13 | Exp. Name: ***Program to convert meters to yards*** | Date: 2023-09-14 |
|---|---|---|

**Aim:**

Write a C Program to convert meters to yards. (formula: 1 yard0.9144 meters)

**Note:** Kindly keep the precision up to 3 digits.

**Source Code:**

yard.c

```
#include <stdio.h>
#define METER_TO_YARD  0.9144
void main() {
        float meter;
        float yard;
        printf("Enter the value in meters: ");
        scanf("%f", &meter);
        yard = meter / METER_TO_YARD;
        printf("%.3f meter : %.3f yard\n", meter, yard);
}
```

## Execution Results - All test cases have succeeded!

| Test Case - 1 |
|---|
| **User Output** |
| Enter the value in meters: |
| 12 |
| 12.000 meter : 13.123 yard |

| Test Case - 2 |
|---|
| **User Output** |
| Enter the value in meters: |
| 425 |
| 425.000 meter : 464.786 yard |

| S.No: 14 | Exp. Name: ***Program to convert temperature from Celsius to Fahrenheit.*** | Date: 2023-09-14 |
|---|---|---|



**Aim:**

Write a C - Programto convert temperature from Celsius to Fahrenheit.

**Source Code:**

conversion.c

```
#include <stdio.h>
void main() {
        float celsius, fahrenheit;
        printf("Enter temperature in celsius: ");
        scanf("%f", &celsius);
        fahrenheit = (celsius * 9 / 5) + 32;
        printf("%.2f Celsius = %.2f Fahrenheit\n", celsius, fahrenheit);
}
```

## Execution Results - All test cases have succeeded!

| Test Case - 1 |
|---|
| **User Output** |
| Enter temperature in celsius: |
| 10 |
| 10.00 Celsius = 50.00 Fahrenheit |

| Test Case - 2 |
|---|
| **User Output** |
| Enter temperature in celsius: |
| 530 |
| 530.00 Celsius = 986.00 Fahrenheit |

| S.No: 15 | Exp. Name: ***Program to swap given two numbers*** | Date: 2023-09-14 |
|---|---|---|

**Aim:**
Write a C Program to swap given two numbers.

**Logic :** By using a temporary variable temp
temp — a,
a — b,
b — temp.

**Source Code:**

temp.c

```
#include <stdio.h>
void main() {
        int x, y, temp;
        printf("Enter the values of x and y: ");
        scanf("%d %d", &x, &y);
        printf("Before swapping x = %d, y = %d\n", x, y);
        temp = x;
        x = y;
        y = temp;
        printf("After swapping x = %d, b = %d\n", x, y);
}
```

## Execution Results - All test cases have succeeded!

| Test Case - 1 |
|---|
| **User Output** |
| Enter the values of x and y: |
| 10 20 |
| Before swapping x = 10, y = 20 |
| After swapping x = 20, b = 10 |

| Test Case - 2 |
|---|
| **User Output** |
| Enter the values of x and y: |
| 247 692 |
| Before swapping x = 247, y = 692 |
| After swapping x = 692, b = 247 |

| S.No: 16 | Exp. Name: ***Program to find the NetSalary of an employee when Basic Salary and Deductions are given.*** | Date: 2023-09-14 |
|---|---|---|



**Aim:**

Write a C Program to find the net salary of an employee when Basic Salary and Deductions aregiven.

**Logic:**By using the below calculations.
DA — 75% of Basic Salary HRA — 15% of Basic Salary
Net Salary — Basic Salary + DA + HRA - Deductions

**Source Code:**

salary.c

```
#include <stdio.h>
void main() {
        float basic, da, hra, dedu;
        float net_salary;
        printf("Enter basic salary: ");
        scanf("%f", &basic);
        printf("Enter deductions: ");
        scanf("%f", &dedu);
        da = (basic * 75) / 100;
        hra = (basic * 15) / 100;
        net_salary = basic + da + hra - dedu;
        printf("Net salary is: %.02f\n", net_salary);
}
```

## Execution Results - All test cases have succeeded!

| Test Case - 1 |
|---|
| **User Output** |
| Enter basic salary: |
| 30000 |
| Enter deductions: |
| 2000 |
| Net salary is: 55000.00 |

| Test Case - 2 |
|---|
| **User Output** |
| Enter basic salary: |
| 1250000 |
| Enter deductions: |
| 10000 |
| Net salary is: 2365000.00 |

| S.No: 17 | Exp. Name: ***Program to calculate the area of a triangle when three sides a,b,c are given.*** | Date: 2023-09-14 |
|---|---|---|





**Aim:**

Write a C Program to calculate the area of a triangle when three sides a,b,c are given.

**Logic:**s = ( a + b + c)/2
Area = sqrt (s* (s- a)* (s - b)* (s- c) )

**Hint:**

To find the square root of a number in the C-Language we should use predefined **math.h** header file, as follow
**# include <math.h>**

ones header file is included you can use predefined functions from math.h header file and to use sqrt() we can do as follow

**Example:**

```
#include <stdio.h>
#include <math.h>
void main()
{
int input=25;
result= sqrt(input);
printf(" The squre root of 25= %d", result);
}
```

**Source Code:**

area.c

```
#include <stdio.h>
#include <math.h>
void main() {
        int s, a, b, c, area;
        printf("Enter the values of a, b and c: ");
        scanf("%d %d %d", &a, &b, &c);
        s = (a + b + c) / 2;
        area = sqrt(s * (s - a) * (s - b) * (s - c));
        printf("Area of triangle: %d\n", area);
}
```

## Execution Results - All test cases have succeeded!

| Test Case - 1 |
|---|
| **User Output** |
| Enter the values of a, b and c: |
| 12 18 8 |
| Area of triangle: 38 |

| Test Case - 2 |
|---|

| User Output |
| --- |
| Enter the values of a, b and c: |
| 56 87 39 |
| Area of triangle: 813 |

| S.No: 18 | Exp. Name: ***Program to check that the volume and surface area of a sphere of radius*** | Date: 2023-09-14 |
|---|---|---|









**Aim:**

Write a program to check that the volume and surface area of a sphere of radius 0.1234567789e2 is approximately 7881.948508and 1915.313445respectively. The volume of a sphere is four-thirdsPi times the radius cubed and the area is four Pi times the radius squared.
Take Pi to be3.14159265.

**Source Code:**

volume.c

```
#include <stdio.h>
#include <math.h>
void main() {
        float r, v, a;
        float pi = 3.14159265;
        printf("Radius of sphere: ");
        scanf("%f", &r);
        v = 4 * pi * r * r * r / 3;
        printf("Volume of sphere: %f\n", v);
        a = 4 * pi * r * r;
        printf("Surface area of sphere: %f\n", a);
}
```

## Execution Results - All test cases have succeeded!

| Test Case - 1 |
|---|
| **User Output** |
| Radius of sphere: |
| 7881.948508 |
| Volume of sphere: 2051112173568.000000 |
| Surface area of sphere: 780687232.000000 |

| Test Case - 2 |
|---|
| **User Output** |
| Radius of sphere: |
| 1915.313445 |
| Volume of sphere: 29431216128.000000 |
| Surface area of sphere: 46098796.000000 |

| S.No: 19 | Exp. Name: ***Program on post increment and pre increment operators ( i++ , ++i )*** | Date: 2023-09-14 |
|---|---|---|



**Aim:**

Write a C program on post increment and pre increment operators (i++ , ++i)

**Source Code:**

1.c

```
#include <stdio.h>
void main() {
        int c;
        printf("Enter a value: ");
        scanf("%d", &c);
        printf("Post increment: %d\n", c++);
        printf("Pre increment: %d\n", ++c);
}
```

## Execution Results - All test cases have succeeded!

| Test Case - 1 |
|---|
| **User Output** |
| Enter a value: |
| 2 |
| Post increment: 2 |
| Pre increment: 4 |

| Test Case - 2 |
|---|
| **User Output** |
| Enter a value: |
| 189 |
| Post increment: 189 |
| Pre increment: 191 |

| S.No: 20 | Exp. Name: ***Program on pre decrement and post decrement operators ( --i , i-- )*** | Date: 2023-09-14 |
|---|---|---|

**Aim:**

Write a C program on pre decrement and post decrement operators ( --i , i-- )

**Source Code:**

1.c

```
#include <stdio.h>
void main() {
        int a, b;
        printf("Enter a value: ");
        scanf("%d", &a);
        printf("Enter b value: ");
        scanf("%d", &b);
        printf("Value of a: %d\n", a);
        printf("Pre decrement of a: %d\n", --a);
        printf("Value of a after decrement: %d\n",a);
        printf("Value of b: %d\n", b);
        printf("Post decrement of b: %d\n", b--);
        printf("Value of b after decrement: %d\n", b);
}
```

## Execution Results - All test cases have succeeded!

| Test Case - 1 |
|---|
| **User Output** |
| Enter a value: |
| 18 |
| Enter b value: |
| 10 |
| Value of a: 18 |
| Pre decrement of a: 17 |
| Value of a after decrement: 17 |
| Value of b: 10 |
| Post decrement of b: 10 |
| Value of b after decrement: 9 |

| Test Case - 2 |
|---|
| **User Output** |
| Enter a value: |
| 357 |
| Enter b value: |
| 518 |
| Value of a: 357 |
| Pre decrement of a: 356 |

```
Value of a after decrement: 356
Value of b: 518
Post decrement of b: 518
Value of b after decrement: 517
```

| S.No: 21 | Exp. Name: ***Program on logical AND operator ( && )*** | Date: 2023-09-14 |
|---|---|---|

**Aim:**

Write a C program on logical AND operator ( && )

**Source Code:**

1.c

```
#include <stdio.h>
void main() {
        int a, b;
        printf("0-FALSE 1-TRUE\n");
        printf("Enter a value: ");
        scanf("%d", &a);
        printf("Enter b value: ");
        scanf("%d", &b);
        printf("AND operator result of a&&b: %d\n", a&&b);
}
```

## Execution Results - All test cases have succeeded!

| Test Case - 1 |
|---|
| **User Output** |
| 0-FALSE 1-TRUE |
| Enter a value: |
| 1 |
| Enter b value: |
| 0 |
| AND operator result of a&&b: 0 |

| Test Case - 2 |
|---|
| **User Output** |
| 0-FALSE 1-TRUE |
| Enter a value: |
| 0 |
| Enter b value: |
| 0 |
| AND operator result of a&&b: 0 |

| S.No: 22 | Exp. Name: ***Program on logical OR operator ( \|\| )*** | Date: 2023-09-14 |
|---|---|---|

**Aim:**

Write a C program on logical OR operator ( || )

**Source Code:**

1.c

```
#include <stdio.h>
void main() {
        int a, b;
        printf("0-FALSE 1-TRUE\n");
        printf("Enter a value: ");
        scanf("%d", &a);
        printf("Enter b value: ");
        scanf("%d", &b);
        printf("OR operator result of a||b: %d\n", a||b);
}
```

## Execution Results - All test cases have succeeded!

| Test Case - 1 |
|---|
| **User Output** |
| 0-FALSE 1-TRUE |
| Enter a value: |
| 1 |
| Enter b value: |
| 0 |
| OR operator result of a\|\|b: 1 |

| Test Case - 2 |
|---|
| **User Output** |
| 0-FALSE 1-TRUE |
| Enter a value: |
| 0 |
| Enter b value: |
| 0 |
| OR operator result of a\|\|b: 0 |

| S.No: 23 | Exp. Name: ***Program on logical NOT operator ( ! )*** | Date: 2023-09-14 |
|---|---|---|

**Aim:**

Write a C program on logical NOT operator ( ! )

**Source Code:**

1.c

```
#include <stdio.h>
#include <conio.h>
void main() {
        int a;
        printf("0-FALSE 1-TRUE\n");
        printf("Enter a value: ");
        scanf("%d", &a);
        printf("NOT operator result is: %d\n", !a);
}
```

## Execution Results - All test cases have succeeded!

| Test Case - 1 |
|---|
| **User Output** |
| 0-FALSE 1-TRUE |
| Enter a value: |
| 0 |
| NOT operator result is: 1 |

| S.No: 24 | Exp. Name: ***Program demonstrating size of operator*** | Date: 2023-09-14 |
|---|---|---|

**Aim:**

Write a C program to demonstrate the size of operator. ( char, short, int, long, float, double, long double, long long )

**Source Code:**

1.c

```
#include <stdio.h>
#include <conio.h>
void main() {
        int number;
        printf("Available cases are\n");
        printf("1.Char\n");
        printf("2.Short\n");
        printf("3.int\n");
        printf("4.long\n");
        printf("5.float\n");
        printf("6.double\n");
        printf("7.long double\n");
        printf("8.long long\n");
        printf("Select a case: ");
        scanf("%d", &number);
        switch(number) {
                case 1:
                        printf("Size Of char: %ld\n", sizeof(char));
                        break;
                case 2:
                        printf("Size Of short: %ld\n", sizeof(short));
                        break;
                case 3:
                        printf("Size Of int: %ld\n", sizeof(int));
                        break;
                case 4:
                        printf("Size Of long: %ld\n", sizeof(long));
                        break;
                case 5:
                        printf("Size Of float: %ld\n", sizeof(float));
                        break;
                case 6:
                        printf("Size Of double: %ld\n", sizeof(double));
                        break;
                case 7:
                        printf("Size Of long double: %ld\n", sizeof(long double));
                        break;
                case 8:
                        printf("Size Of long long: %ld\n", sizeof(long long));
                        break;
                default:
                        printf("Select the correct case\n");
        }
}
```

## Execution Results - All test cases have succeeded!

| Test Case - 1 |
|---|
| **User Output** |
| Available cases are |
| 1.Char |
| 2.Short |
| 3.int |
| 4.long |
| 5.float |
| 6.double |
| 7.long double |
| 8.long long |
| Select a case: |
| 1 |
| Size Of char: 1 |

| Test Case - 2 |
|---|
| **User Output** |
| Available cases are |
| 1.Char |
| 2.Short |
| 3.int |
| 4.long |
| 5.float |
| 6.double |
| 7.long double |
| 8.long long |
| Select a case: |
| 3 |
| Size Of int: 4 |

| Test Case - 3 |
|---|
| **User Output** |
| Available cases are |
| 1.Char |
| 2.Short |
| 3.int |
| 4.long |
| 5.float |
| 6.double |
| 7.long double |
| 8.long long |
| Select a case: |
| 5 |
| Size Of float: 4 |

| Test Case - 4 |
| --- |
| **User Output** |
| Available cases are |
| 1.Char |
| 2.Short |
| 3.int |
| 4.long |
| 5.float |
| 6.double |
| 7.long double |
| 8.long long |
| Select a case: |
| 7 |
| Size Of long double: 16 |

| Test Case - 5 |
| --- |
| **User Output** |
| Available cases are |
| 1.Char |
| 2.Short |
| 3.int |
| 4.long |
| 5.float |
| 6.double |
| 7.long double |
| 8.long long |
| Select a case: |
| 9 |
| Select the correct case |

| Test Case - 6 |
| --- |
| **User Output** |
| Available cases are |
| 1.Char |
| 2.Short |
| 3.int |
| 4.long |
| 5.float |
| 6.double |
| 7.long double |
| 8.long long |
| Select a case: |
| 8 |
| Size Of long long: 8 |

| S.No: 25 | Exp. Name: ***Program to demonstrate the use of relational and all bitwise operators. ( &, \|, ^, ~, <<, >> )*** | Date: 2023-09-14 |
|---|---|---|









**Aim:**

Write a C program to demonstrate the use of relational and all bitwise operators. ( & , | , ^ , ~ , << , >> )

**Source Code:**

1.c

```
#include <stdio.h>
#include <conio.h>
void main() {
        int a, b, c;
        printf("Enter a value: ");
        scanf("%d", &a);
        printf("Enter b value: ");
        scanf("%d", &b);
        c = a & b;
        printf("Value of a&b is: %d\n", c);
        c = a | b;
        printf("Value of a|b is: %d\n", c);
        c = a ^ b;
        printf("Value of a^b is: %d\n", c);
        c =~ a;
        printf("Value of ~a is: %d\n", c);
        c = a << b;
        printf("Value of a<<b is: %d\n", c);
        c = a >> b;
        printf("Value of a>>b is: %d\n", c);
}
```

## Execution Results - All test cases have succeeded!

| Test Case - 1 |
|---|
| **User Output** |
| Enter a value: |
| 60 |
| Enter b value: |
| 13 |
| Value of a&b is: 12 |
| Value of a\|b is: 61 |
| Value of a^b is: 49 |
| Value of ~a is: -61 |
| Value of a< |
| **Value of a>>b is: 0** |

| Test Case - 2 |
|---|

| User Output |
| --- |
| Enter a value: |
| 329 |
| Enter b value: |
| 246 |
| Value of a&b is: 64 |
| Value of a\|b is: 511 |
| Value of a^b is: 447 |
| Value of ~a is: -330 |
| Value of a< |
| Value of a>>b is: 0 |

| S.No: 26 | Exp. Name: ***Program to find distance, when initial velocity, time, acceleration are taken as user input*** | Date: 2023-09-14 |
|---|---|---|









**Aim:**

**Write a program to find the value of s, the distance traveled by an object $s=u*t+0.5*a*t^2$. Where u is the initial velocity, t is the time and a is the acceleration.**

**Source Code:**

1.c

```
#include <stdio.h>
#include <math.h>
void main() {
        float u, t, a, s, z;
        printf("Enter value of initial velocity: ");
        scanf("%f", &u);
        printf("Enter the time: ");
        scanf("%f", &t);
        printf("Enter value of acceleration: ");
        scanf("%f", &a);
        z = t * t;
        s = u * t + 0.5 * a * z;
        printf("The distance travelled by the object is: %f\n", s);
}
```

## Execution Results - All test cases have succeeded!

| Test Case - 1 |
|---|
| **User Output** |
| Enter value of initial velocity: |
| **123.4** |
| Enter the time: |
| **34** |
| Enter value of acceleration: |
| **234.8** |
| The distance travelled by the object is: 139910.000000 |

| Test Case - 2 |
|---|
| **User Output** |
| Enter value of initial velocity: |
| **123** |
| Enter the time: |
| **56** |
| Enter value of acceleration: |
| **78** |
| The distance travelled by the object is: 129192.000000 |

| S.No: 27 | Exp. Name: ***Program to find the given number is even or odd using a conditional operator*** | Date: 2023-09-14 |
|---|---|---|



**Aim:**

**Write a program to find the given number is even or odd using a conditional operator.**

**Source Code:**

1.c

```
#include <stdio.h>
void main() {
        int n;
        printf("Enter an integer: ");
        scanf("%d",&n);
        n % 2 == 0 ? printf("The given number is even number\n") : printf("The given number
is odd number\n");
}
```

## Execution Results - All test cases have succeeded!

| Test Case - 1 |
|---|
| **User Output** |
| Enter an integer: |
| **45** |
| The given number is odd number |

| Test Case - 2 |
|---|
| **User Output** |
| Enter an integer: |
| **774** |
| The given number is even number |

| S.No: 28 | Exp. Name: ***Program to find the biggest of three numbers using a conditional operator.*** | Date: 2023-09-14 |
|---|---|---|

**Aim:**

**Write a program to find the biggest of three numbers using a conditional operator.**

**Source Code:**

1.c

```
#include <stdio.h>
void main() {
        int a, b, c, big;
        printf("Enter three numbers: ");
        scanf("%d %d %d", &a, &b, &c);
        big = a > b ? (a > c ? a : c) : (b > c ? b : c);
        printf("The biggest number is: %d\n", big);
}
```

## Execution Results - All test cases have succeeded!

| Test Case - 1 |
|---|
| **User Output** |
| Enter three numbers: |
| **12 5 18** |
| The biggest number is: 18 |

| Test Case - 2 |
|---|
| **User Output** |
| Enter three numbers: |
| **254 894 752** |
| The biggest number is: 894 |

| S.No: 29 | Exp. Name: ***Program to find out the value of (a+b)²*** | Date: 2023-09-14 |
|---|---|---|

**Aim:**

**Write a program to find out the value of (a+b)²**

**Source Code:**

1.c

```
#include <stdio.h>
int main(void) {
        float a, b, x;
        printf("Enter value of a: ");
        scanf("%f", &a);
        printf("Enter value of b: ");
        scanf("%f", &b);
        x = (a * a) + (b * b) + (2 * a * b);
        printf("Result: %f\n", x);
        return 0;
}
```

## Execution Results - All test cases have succeeded!

| Test Case - 1 |
|---|
| **User Output** |
| Enter value of a: |
| **1** |
| Enter value of b: |
| **2** |
| Result: 9.000000 |

| Test Case - 2 |
|---|
| **User Output** |
| Enter value of a: |
| **219** |
| Enter value of b: |
| **96** |
| Result: 99225.000000 |

| S.No: 30 | Exp. Name: ***Program demonstrating conditional operator. ( ? : )*** | Date: 2023-09-14 |
|---|---|---|

**Aim:**

**Write a C program to demonstrate conditional operator. ( ? : )**

**Syntax:**

**( variable = expression1 ? expression2 : expression 3)**

**Source Code:**

1.c

```
#include <stdio.h>
void main() {
        int a, b, big;
        printf("Enter two numbers: ");
        scanf("%d%d", &a, &b);
        printf("a = %d, b = %d\n",a,b);
        big = a > b ? a : b;
        printf("Largest number: %d\n", big);
}
```

## Execution Results - All test cases have succeeded!

| Test Case - 1 |
|---|
| **User Output** |
| Enter two numbers: |
| **18 36** |
| a = 18, b = 36 |
| Largest number: 36 |

| Test Case - 2 |
|---|
| **User Output** |
| Enter two numbers: |
| **318 996** |
| a = 318, b = 996 |
| Largest number: 996 |

| S.No: 31 | Exp. Name: ***Program Using Conditional Control Statements(if,if-else,nested if).*** | Date: 2023-09-14 |
|---|---|---|









**Aim:**

**Write a C program to print the Number entered by the user only if the number entered is Positive.**

**ALGORITHM:**

**Step 1: Start**
**Step 2: Input num**
**Step 3: Check if(num>0) then**
**Step 3.1: Print "number is positive"**
**else**
**Step 3.2: Print "number is negative"**
**Step 4: Stop**

**Source Code:**

c.c

```
#include <stdio.h>
void main(){
        int num;
        printf("Enter a number to check: ");
        scanf("%d", &num);
        if (num > 0) {
                printf("Number is positive\n");
        }
        else {
                printf("Number is negative\n");
        }
}
```

## Execution Results - All test cases have succeeded!

| Test Case - 1 |
|---|
| **User Output** |
| Enter a number to check: |
| **3547** |
| Number is positive |

| Test Case - 2 |
|---|
| **User Output** |
| Enter a number to check: |
| **-3147** |
| Number is negative |

| S.No: 32 | Exp. Name: ***Program to check whether a given number is even or odd.*** | Date: 2023-09-14 |
|---|---|---|



**Aim:**

**Write a C program to check whether a given number is EVEN or ODD.**

**ALGORITHM:**

**Step 1: Start**
**Step 2: Input num1**
**Step 3: Check if(num1 is divisible by 2) then**
**Step 3.1: Print "number is even"**
**else**
**Step 3.2: Print "number is odd"**
**Step 4: Stop**

**Source Code:**

evenOrOdd.c

```
#include <stdio.h>
void main() {
        int num;
        printf("Enter a number to check: ");
        scanf("%d", &num);
        if ((num % 2) == 0) {
                printf("%d is even\n", num);
        }
        else {
                printf("%d is odd\n", num);
        }
}
```

Execution Results - All test cases have succeeded!

| Test Case - 1 |
|---|
| **User Output** |
| Enter a number to check: |
| **30** |
| 30 is even |

| Test Case - 2 |
|---|
| **User Output** |
| Enter a number to check: |
| **47** |
| 47 is odd |

| S.No: 33 | Exp. Name: ***Program to check whether a given number is multiple of 5 or not.*** | Date: 2023-09-14 |
|---|---|---|



**Aim:**

**Write a C program to check whether a given number is multiple of 5 or not.**

**ALGORITHM:**

**Step 1: Start**
**Step 2: Input num1**
**Step 3: Check if(num1 is divisible by 5) then**
**Step 3.1: Print "number is a multiple of 5"**
**else**
**Step 3.2: Print "number is not a multiple of 5"**
**Step 4: Stop**

**Source Code:**

multiple.c

```
#include <stdio.h>
void main() {
        int num;
        printf("Enter a number to check: ");
        scanf("%d", &num);
        if ((num % 5) == 0) {
                printf("%d is a multiple of 5\n", num);
        }
        else {
                printf("%d is not a multiple of 5\n", num);
        }
}
```

Execution Results - All test cases have succeeded!

| Test Case - 1 |
|---|
| **User Output** |
| Enter a number to check: |
| **20** |
| 20 is a multiple of 5 |

| Test Case - 2 |
|---|
| **User Output** |
| Enter a number to check: |
| **67** |
| 67 is not a multiple of 5 |

| S.No: 34 | Exp. Name: ***Program to find the biggest of 3 numbers.*** | Date: 2023-09-14 |
|---|---|---|





**Aim:**

**Write a C program to find the Biggest of 3 numbers.**

**ALGORITHM:**

**Step 1: Start**
**Step 2: Input a,b,c**
**Step 3: Check if(a>b and a>c) then**
**Step 3.1: Print "a is greatest"**
**else**
**Step 4: Check if(b>a and b>c) then**
**Step 4.1: Print "b is greatest"**
**else**
**Step 5: Check if(c>a and c>b) then**
**Step 5.1: Print "c is greatest"**
**else**
**Step 6: Stop**

**Source Code:**

largest.c

```
#include <stdio.h>
void main(){
        int num1, num2, num3;
        printf("Number 1: ");
        scanf("%d", &num1);
        printf("Number 2: ");
        scanf("%d", &num2);
        printf("Number 3: ");
        scanf("%d", &num3);
        if (num1 > num2) {
                if (num1 > num3) {
                        printf("%d is greater among three\n", num1);
                }
                else {
                        printf("%d is greater among three\n", num3);
                }
        }
        else if (num2 > num3)
                printf("%d is greater among three\n", num2);
        else
                printf("%d is greater among three\n", num3);
}
```





## Execution Results - All test cases have succeeded!

| Test Case - 1 |
|---|
| **User Output** |
| Number 1: |
| **10** |

| Number 2: |
| --- |
| **11** |
| Number 3: |
| **12** |
| 12 is greater among three |

| Test Case - 2 |
| --- |
| **User Output** |
| Number 1: |
| **58** |
| Number 2: |
| **96** |
| Number 3: |
| **42** |
| 96 is greater among three |

| Test Case - 3 |
| --- |
| **User Output** |
| Number 1: |
| **89** |
| Number 2: |
| **27** |
| Number 3: |
| **52** |
| 89 is greater among three |

| S.No: 35 | Exp. Name: ***Program to find the nature of the roots and roots of a Quadratic equation (ax2+bx+c=0).*** | Date: 2023-09-14 |
|---|---|---|









**Aim:**

**Write a C program to find the nature of the roots and roots of a Quadratic equation ($ax^2+bx+c=0$).**

**$d=b^2-4ac$**
**If d=0, roots (r1, r2) are real and equal, r1=r2=-b/2a**
**If d>0, roots (r1, r2) are real, r1= (-b + sqrt (d))/2a, r2=(-b-sqrt(d))/2a**
**Ifd<0,roots(r1,r2)arecomplex,r1=(-b)/2a+i(sqrt(-d))/2a,r2=(-b)/2a-i(sqrt(-d))/2a**

**Source Code:**

c.c

```
#include <math.h>
#include <stdio.h>
int main() {
    double a, b, c, discriminant, root1, root2, realPart, imagPart;
    printf("Enter coefficients a: ");
    scanf("%lf", &a);
    printf("Enter coefficients b: ");
    scanf("%lf", &b);
    printf("Enter coefficients c: ");
    scanf("%lf", &c);

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

    // condition for real and different roots
    if (discriminant > 0) {
        root1 = (-b + sqrt(discriminant)) / (2 * a);
        root2 = (-b - sqrt(discriminant)) / (2 * a);
        printf("root1 = %.2lf and root2 = %.2lf", root1, root2);
    }

    // condition for real and equal roots
    else if (discriminant == 0) {
        root1 = root2 = -b / (2 * a);
        printf("root1 = root2 = %.2lf;", root1);
    }

    // if roots are not real
    else {
        realPart = -b / (2 * a);
        imagPart = sqrt(-discriminant) / (2 * a);
        printf("root1 = %.2lf+%.2lfi and root2 = %.2f-%.2fi", realPart, imagPart, realPart,
imagPart);
    }

    return 0;
}
```

## Execution Results - All test cases have succeeded!

| Test Case - 1 |
| --- |
| **User Output** |
| Enter coefficients a: |
| **2** |
| Enter coefficients b: |
| **-11** |
| Enter coefficients c: |
| **5** |
| root1 = 5.00 and root2 = 0.50 |

| Test Case - 2 |
| --- |
| **User Output** |
| Enter coefficients a: |
| **1** |
| Enter coefficients b: |
| **-7** |
| Enter coefficients c: |
| **12** |
| root1 = 4.00 and root2 = 3.00 |

| Test Case - 3 |
| --- |
| **User Output** |
| Enter coefficients a: |
| **4** |
| Enter coefficients b: |
| **1** |
| Enter coefficients c: |
| **6** |
| root1 = -0.12+1.22i and root2 = -0.12-1.22i |

| S.No: 36 | Exp. Name: ***Program to calculate value of f(x) if x has different ranges of given values.*** | Date: 2023-09-14 |
|---|---|---|









**Aim:**

**Write a C program to calculate the value of f(x) if x has different ranges of values as below.**

**$f(x) = x^2 + 2$ if 0<=x<=10**
**$f(x) = x^2 + 2x$ if 11<=x<=20**
**$f(x) = x^3 + 2x^2$ if 21<=x<=30**
**f(x) = 0 if x > 30**

**Source Code:**

c.c

```
#include <math.h>
#include <stdio.h>
int main() {
        int x,a,b;
        printf("Enter x Value:"); scanf("%d",&x);
        switch(x){
                case 0 ... 10:{
                        b=x*x+2;
                        printf("Value of F(x)= %d",b);
                        break;
                }
                case 11 ... 20:{
                        b=x*x+2*x;
                        printf("Value of F(x)= %d",b);
                        break;
                }
                case 21 ... 30:{
                        b=x*x*x+2*x*x;
                        printf("Value of F(x)= %d",b);
                        break;
                }
                default:{
                        printf("Value of F(x)= 0");
                }
        }
        return 0;
}
```

## Execution Results - All test cases have succeeded!

| Test Case - 1 |
|---|
| **User Output** |
| Enter x Value: |
| **3** |
| Value of F(x)= 11 |

| Test Case - 2 |
| --- |
| **User Output** |
| `Enter x Value:` |
| **11** |
| `Value of F(x)= 143` |

| Test Case - 3 |
| --- |
| **User Output** |
| `Enter x Value:` |
| **25** |
| `Value of F(x)= 16875` |

| Test Case - 4 |
| --- |
| **User Output** |
| `Enter x Value:` |
| **33** |
| `Value of F(x)= 0` |

| S.No: 37 | Exp. Name: ***Program to compute the net amount to be paid as per the data provided in the table.*** | Date: 2023-09-14 |
|---|---|---|



**Aim:**

**Program using switch and if statements to compute the Net Amount to be paid by the customer to the cloth show room based on Purchase Amount and Discount.**

| Purchase Amount | Discount | |
|---|---|---|
| | Mill Cloth | Handloom Items |
| 0-100 | -- | 5% |
| 101-200 | 5% | 7.5% |
| 201-300 | 7.5% | 10% |
| Above 300 | 10% | 15% |

**Source Code:**

```
c.c
```

```c
#include <math.h>
#include <stdio.h>
int main() {
        int x,y; float a,b,z;
        printf("Enter Value:"); scanf("%d",&x);
        switch(x){
                case 0 ... 100:{
                        a=x;
                        printf("Mill Cloth Value= %.2f\n",a);
                        b=x-5*x/100.00;
                        printf("Handloom Items Value= %.2f",b);
                        break;
                }
                case 101 ... 200:{
                        a=x-5*x/100.00;
                        printf("Mill Cloth Value= %.2f\n",a);
                        b=x-7.5*x/100.00;
                        printf("Handloom Items Value= %.2f",b);
                        break;
                }
                case 201 ... 300:{
                        a=x-7.5*x/100.00;
                        printf("Mill Cloth Value= %.2f\n",a);
                        b=x-10*x/100.00;
                        printf("Handloom Items Value= %.2f",b);
                        break;
                }
                default:{
                        a=x-10*x/100.00;
                        printf("Mill Cloth Value= %.2f\n",a);
                        b=x-15*x/100.00;
                        printf("Handloom Items Value= %.2f",b);
                        break;
                }
        }
        return 0;
}
```

## Execution Results - All test cases have succeeded!

| Test Case - 1 |
|---|
| **User Output** |
| Enter Value: |
| **350** |
| Mill Cloth Value= 315.00 |
| Handloom Items Value= 297.50 |

| Test Case - 2 |
|---|
| **User Output** |
| Enter Value: |

| 100 |
| --- |
| Mill Cloth Value= 100.00 |
| Handloom Items Value= 95.00 |

| Test Case - 3 |
| --- |
| **User Output** |
| Enter Value: |
| **300** |
| Mill Cloth Value= 277.50 |
| Handloom Items Value= 270.00 |

| Test Case - 4 |
| --- |
| **User Output** |
| Enter Value: |
| **200** |
| Mill Cloth Value= 190.00 |
| Handloom Items Value= 185.00 |

| S.No: 38 | Exp. Name: ***Program to read two integer values and a basic operator from the keyboard and based on that operator performs the specified operation.*** | Date: 2023-11-10 |
|---|---|---|









**Aim:**

**Write a C program to read two integer values and a basic operator from the keyboard and based onthat operator performs the specified operation.**

ALGORITHM:

```
Step 1:   Start
Step 2:   Input a, b, operator (any one of +,-,*,/)
Step 3:   switch (operator)
          {
                case  '+':
                      result←a+b
                      break;
                case  '-':
                      result←a-b
                      break;
                case  '*':
                      result←a*b
                      break;
                case  '/':
                      result←a/b
                      break;
                default:
                      Display  "Invalid Operator"
                      break;
          }
Step 3:   Display result
```

**Source Code:**

Operation.c

```c
#include <stdio.h>

int main() {
    char operator;
    int num1, num2;

    printf("Enter an operator: ");
    scanf(" %c", &operator);

    printf("Enter two integer values: ");
    scanf("%d %d", &num1, &num2);

    switch (operator) {
        case '+':
            printf("Result: %d\n", num1 + num2);
            break;
        case '-':
            printf("Result: %d\n", num1 - num2);
            break;
        case '*':
            printf("Result: %d\n", num1 * num2);
            break;
        case '/':
            if (num2 != 0) {
                printf("Result: %d\n", num1 / num2);
            } else {
                printf("Division by zero is not allowed.\n");
            }
            break;
        default:
            printf("Invalid operator\n");
    }

    return 0;
}
```

## Execution Results - All test cases have succeeded!

| Test Case - 1 |
|---|
| **User Output** |
| Enter an operator: |
| **+** |
| Enter two integer values: |
| **20 10** |
| Result: 30 |

| Test Case - 2 |
|---|
| **User Output** |
| Enter an operator: |
| ***** |

| Enter two integer values: |
|---|
| **60 16** |
| Result: 960 |

| Test Case - 3 |
|---|
| **User Output** |
| Enter an operator: |
| **#** |
| Enter two integer values: |
| **20 5** |
| Invalid operator |

| S.No: 39 | Exp. Name: ***Program to execute a goto statement..?*** | Date: 2023-11-10 |
|---|---|---|









**Aim:**

**Write a C program to execute a goto statement by following the given algorithm**

**ALGORITHM:**

**Step 1: Input a number**
**Step 2: b = number+5**
**Step 3: c = number+10**
**Step 4: declare LOOP:do**
**Step 5: Check if (number==b)**
**Step 5.1: a = a + 1**
**Step 5.2: goto LOOP**
**Step 6: print number**
**Step 7: a = a+1**
**Step 8: while(number < c)**
**Step 9: Stop.**

**Source Code:**

goto.c

```
#include <stdio.h>
void main() {
        int a, b, c;
        printf("Enter a value: ");
        scanf("%d", &a);
        b = a + 5;
        c = a + 10;
        LOOP:do{
                if (a == b) {
                        a = a + 1;
                        goto LOOP;
                }
                printf("Value of a: %d\n", a);
                a++;
        } while( a < c);
}
```

## Execution Results - All test cases have succeeded!

| Test Case - 1 |
|---|
| **User Output** |
| Enter a value: |
| **10** |
| Value of a: 10 |
| Value of a: 11 |
| Value of a: 12 |
| Value of a: 13 |
| Value of a: 14 |
| Value of a: 16 |

| Value of a: 17 |
| --- |
| Value of a: 18 |
| Value of a: 19 |

| Test Case - 2 |
| --- |
| **User Output** |
| Enter a value: |
| **253** |
| Value of a: 253 |
| Value of a: 254 |
| Value of a: 255 |
| Value of a: 256 |
| Value of a: 257 |
| Value of a: 259 |
| Value of a: 260 |
| Value of a: 261 |
| Value of a: 262 |

| S.No: 40 | Exp. Name: ***Program to print the student grade based on average marks.*** | Date: 2023-11-10 |
|---|---|---|









**Aim:**

**Write a C program to print the student grade based on average marks: If avg>=70 grade=Distinction , if avg>=60 but <70grade="First class", If avg>45 but<60 grade="Second class" otherwise grade="Fail".**

**Grades should be any one of the following**

**Distinction**
**First class**
**Second class**
**Fail**
**Pass**

**ALGORITHM:**

Step 1: Input marks1, marks2, marks3, marks4, marks5
Step 2: avg ← (marks1 + marks2 + marks3 +marks4 +marks5)/5
Step 3: check if (marks1 < 35 or marks2 < 35 ormarks3 < 35 or marks4 < 35 or marks5< 35)
Step 3.1: print "**Fail**"
Step 3.2: exit
Step 4: else if (avg>=70) then
Step 4.1 print "**Distinction**"
Step 5: elseif (avg>=60 and <70) then
Step 5.1 print"**First class**"
Step 6: else if (avg>45 and < 60) then
Step 6.1 print "**secondclass**"
Step 7: else
Step 7.1 print "**Pass**"
Step 8: endif
Step 9: Stop

**Source Code:**

Marks.c

```c
#include <stdio.h>
void main() {
        int m1, m2, m3, m4, m5;
        printf("Enter the 5 subjects marks: ");
        scanf("%d %d %d %d %d", &m1, &m2, &m3, &m4, &m5);
        int avg = (m1 + m2 + m3 + m4 + m5) / 5;
        if (m1 < 35 || m2 < 35 || m3 < 35 || m4 < 35 || m5 < 35) {
                printf("Fail\n");
        }
        else if (avg >= 70) {
                printf("Distinction\n");
        }
        else if (avg >= 60 && avg < 70) {
                printf("First class\n");
        }
        else if (avg > 45 && avg < 60) {
                printf("Second class\n");
        }
        else {
                printf("Pass\n");
        }
}
```

## Execution Results - All test cases have succeeded!

| Test Case - 1 |
|---|
| **User Output** |
| Enter the 5 subjects marks: |
| **92 87 89 78 91** |
| Distinction |

| Test Case - 2 |
|---|
| **User Output** |
| Enter the 5 subjects marks: |
| **68 71 70 55 63** |
| First class |

| Test Case - 3 |
|---|
| **User Output** |
| Enter the 5 subjects marks: |
| **95 45 34 74 54** |
| Fail |

| Test Case - 4 |
|---|
| **User Output** |

```
Enter the 5 subjects marks:
36 36 36 36 38
Pass
```

| S.No: 41 | Exp. Name: ***Program to check whether the given input character is an integer or a character, if it is a character then check whether it is an uppercase or lowercase character.*** | **Date: 2023-11-10** |
|---|---|---|



**Aim:**

**Write a C program to check whether the given input character is an integer or a character, if it is acharacter then check whether it is an uppercase or lowercase character.**

**ALGORITHM:**

**Step 1:Input a character (ch)**
**Step 2:check if (ch >=48 and ch <=57) then**
**Step 2.1:Print"Input character is a digit"**
**Step 3:check else if (ch >=65 and ch <=90) then**
**Step 3.1:Print "Input character is an uppercase character"**
**Step 4:check else if (ch >= 97 and ch <=122)then**
**Step 4.1:Print "Input character is a lowercase character"**
**Step 5:check else**
**Step 5.1:Print "Input character is a special character"**
**Step 6:Stop.**

**Source Code:**

Char.c

```
#include <stdio.h>

int main() {
    char ch;

    printf("Enter a character: ");
    scanf(" %c", &ch);

    if (ch >= '0' && ch <= '9') {
        printf("Input character is digit\n");
    } else if (ch >= 'A' && ch <= 'Z') {
        printf("Input character is an uppercase character\n");
    } else if (ch >= 'a' && ch <= 'z') {
        printf("Input character is a lowercase character\n");
    } else {
        printf("Input character is a special character\n");
    }

    return 0;
}
```

## Execution Results - All test cases have succeeded!

| Test Case - 1 |
|---|
| **User Output** |
| Enter a character: |
| **9** |

```
Input character is digit
```

| Test Case - 2 |
| --- |
| **User Output** |
| `Enter a character:` |
| **&** |
| `Input character is a special character` |

| Test Case - 3 |
| --- |
| **User Output** |
| `Enter a character:` |
| **b** |
| `Input character is a lowercase character` |

| S.No: 42 | Exp. Name: ***Program to calculate the sum and average of given positive numbers.*** | Date: 2023-11-10 |
|---|---|---|









**Aim:**

**Write a C program to calculate the sum and average of given positive numbers. If the user enters a negative number, the sum and average to be displayed(Using goto statement).**

**Source Code:**

sumAndAverage.c

```
#include <stdio.h>

int main() {
    int count = 0;
    double number, sum = 0.0;

input:
    scanf("%lf", &number);

    if (number < 0) {
        double average = sum / count;
        printf("Sum: %.2lf\n", sum);
        printf("Average: %.2lf\n", average);
        return 0;
    } else {
        sum += number;
        count++;
        goto input;
    }

    return 0;
}
```

## Execution Results - All test cases have succeeded!

| Test Case - 1 |
|---|
| **User Output** |
| **3** |
| **4.3** |
| **9.3** |
| **-2.9** |
| Sum: 16.60 |
| Average: 5.53 |

| Test Case - 2 |
|---|
| **User Output** |
| **5** |
| **8** |

| 9.4 |
| --- |
| -5 |
| Sum: 22.40 |
| Average: 7.47 |

| S.No: 43 | Exp. Name: ***program to check vowel or consonant.*** | Date: 2023-11-10 |
|---|---|---|

**Aim:**

**Write a C program to input an alphabet and check whether it is vowel or consonant using switch case.**

**Source Code:**

voc.c

```
#include <stdio.h>

int main() {
    char ch;
    scanf(" %c", &ch); // Note the space before %c to consume any whitespace characters

    switch (ch) {
        case 'a':
        case 'A':
        case 'e':
        case 'E':
        case 'i':
        case 'I':
        case 'o':
        case 'O':
        case 'u':
        case 'U':
            printf("%c is a vowel.\n", ch);
            break;
        default:
            printf("%c is a consonant.\n", ch);
            break;
    }

    return 0;
}
```

## Execution Results - All test cases have succeeded!

| Test Case - 1 |
|---|
| **User Output** |
| **c** |
| c is a consonant. |

| Test Case - 2 |
|---|
| **User Output** |
| **a** |
| a is a vowel. |

| S.No: 44 | Exp. Name: ***Program to print weekday.*** | Date: 2023-11-10 |
|---|---|---|

**Aim:**

**Write a C program to print weekday based on given number using if else ladder statement.**

**Note: Consider 1 to 7 days.**

**Source Code:**

weekday.c

```
#include <stdio.h>

int main() {
    int dayNumber;
    scanf("%d", &dayNumber);

    if (dayNumber >= 1 && dayNumber <= 7) {
        if (dayNumber == 1) {
            printf("Sunday\n");
        } else if (dayNumber == 2) {
            printf("Monday\n");
        } else if (dayNumber == 3) {
            printf("Tuesday\n");
        } else if (dayNumber == 4) {
            printf("Wednesday\n");
        } else if (dayNumber == 5) {
            printf("Thursday\n");
        } else if (dayNumber == 6) {
            printf("Friday\n");
        } else {
            printf("Saturday\n");
        }
    } else {
        printf("Invalid input\n");
    }

    return 0;
}
```

## Execution Results - All test cases have succeeded!

| Test Case - 1 |
|---|
| **User Output** |
| **4** |
| Wednesday |

| Test Case - 2 |
|---|
| **User Output** |
| **9** |

Invalid input

| S.No: 45 | Exp. Name: ***program to calculate gross salary.*** | Date: 2023-11-10 |
|---|---|---|





**Aim:**

**Write a C program to input basic salary of an employee and calculate gross salary according to given conditions.**
**Basic Salary <= 10000 : HRA = 20%, DA = 80%**
**Basic Salary is between 10001 to 20000 : HRA = 25%, DA = 90%**
**Basic Salary >= 20001 : HRA = 30%, DA = 95%**

**Note: The formula for DA and HRA is da = basic_salary * (DA/100)**

**If DA = 80% then the statement becomes da = basic_salary * (80/100).**
**Calculate final gross salary using the formula gross_salary = basic_salary + DA + HRA**

**Source Code:**

salary.c

```
#include <stdio.h>

int main() {
    double basicSalary, grossSalary;
    double HRA, DA;
    scanf("%lf", &basicSalary);

    if (basicSalary <= 10000) {
        HRA = basicSalary * (20.0 / 100.0);
        DA = basicSalary * (80.0 / 100.0);
    } else if (basicSalary > 10001 && basicSalary < 20000) {
        HRA = basicSalary * (25.0 / 100.0);
        DA = basicSalary * (90.0 / 100.0);
    } else {
        HRA = basicSalary * (30.0 / 100.0);
        DA = basicSalary * (95.0 / 100.0);
    }

    // Calculate the gross salary
    grossSalary = basicSalary + HRA + DA;
    printf("Gross Salary: %.2lf\n", grossSalary);

    return 0;
}
```

Execution Results - All test cases have succeeded!

| Test Case - 1 |
|---|
| **User Output** |
| **30000** |
| Gross Salary: 67500.00 |

| Test Case - 2 |
|---|

| User Output |
| --- |
| **10000** |
| Gross Salary: 20000.00 |

| S.No: 46 | Exp. Name: ***Program to calculate electricity bill*** | Date: 2023-11-10 |
|---|---|---|





**Aim:**

**Write a C program to input electricity unit charge and calculate the total electricity bill according to the given condition:**

**For first 50 units RS. 0.50/ unit**
**For next 100 units RS. 0.75/ unit**
**For next 50 units RS. 1.20/ unit**
**For unit above 250 RS. 1.50/ unit**
**An additional surcharge of 20% is added to the bill.**

**Note: After calculating total amount. Calculate the surcharge amount i.e. sur_charge = total_amt * 0.20. Add surcharge amount to net amount. Which is given by net_amt = total_amt _ sur_charge.**

**Source Code:**

electricitybill.c

```
#include <stdio.h>

int main() {
    float unitCharge, totalAmount, surcharge, netAmount;
    scanf("%f", &unitCharge);

    if (unitCharge <= 50) {
        totalAmount = unitCharge * 0.50;
    } else if (unitCharge <= 150) {
        totalAmount = 50 * 0.50 + (unitCharge - 50) * 0.75;
    } else if (unitCharge <= 200) {
        totalAmount = 50 * 0.50 + 100 * 0.75 + (unitCharge - 150) * 1.20;
    } else {
        totalAmount = 50 * 0.50 + 100 * 0.75 + 50 * 1.20 + (unitCharge - 200) * 1.50;
    }

    // Calculate surcharge
    surcharge = totalAmount * 0.20;

    // Calculate net amount
    netAmount = totalAmount + surcharge;
    printf("Electricity Bill: %.2f\n", netAmount);

    return 0;
}
```

## Execution Results - All test cases have succeeded!

| Test Case - 1 |
|---|
| **User Output** |
| **200** |
| Electricity Bill: 192.00 |

| Test Case - 2 |
| --- |
| **User Output** |
| **180** |
| `Electricity Bill: 163.20` |

| S.No: 47 | Exp. Name: ***Program to find the sum of 1 to n integers.*** | Date: 2023-10-13 |
|---|---|---|

**Aim:**

**Write a C Program to find the sum of 1 to n integers.**

**Source Code:**

sumofn.c

```
#include <stdio.h>
void main() {
        int n, i, sum = 0;
        printf("Enter a positive integer: ");
        scanf("%d", &n);
        i = 1;
        while (i <= n) {
                sum += i;
                ++i;
        }
        printf("Sum = %d\n", sum);
}
```

## Execution Results - All test cases have succeeded!

| Test Case - 1 |
|---|
| **User Output** |
| Enter a positive integer: |
| **100** |
| Sum = 5050 |

| Test Case - 2 |
|---|
| **User Output** |
| Enter a positive integer: |
| **239** |
| Sum = 28680 |

| S.No: 48 | Exp. Name: ***Program to find whether the given number is palindrome or not.*** | Date: 2023-10-13 |
|---|---|---|

**Aim:**

**Write a C Program to find whether the given number is palindrome or not.**

**Source Code:**

palindrome.c

```
#include <stdio.h>
void main() {
        int n, s = 0, rem, originalInteger;
        printf("Enter an integer: ");
        scanf("%d", &n);
        originalInteger = n;
        while(n != 0) {
                rem = n % 10;
                s = s * 10 + rem;
                n = n / 10;
        }
        if (originalInteger == s)
                printf("%d is a palindrome\n", originalInteger);
        else
                printf("%d is not a palindrome\n", originalInteger);
}
```

## Execution Results - All test cases have succeeded!

| Test Case - 1 |
|---|
| **User Output** |
| Enter an integer: |
| **44** |
| 44 is a palindrome |

| Test Case - 2 |
|---|
| **User Output** |
| Enter an integer: |
| **23** |
| 23 is not a palindrome |

| S.No: 49 | Exp. Name: ***Program to print fibonacci sequence upto n*** | Date: 2023-10-13 |
|---|---|---|

**Aim:**

**Write a C program to print the Fibonacci sequence upto a given number n.**

**Source Code:**

fibonacci.c

```
#include <stdio.h>
void main() {
        int t1 = 0, t2 = 1, nextTerm = 0, n;
        printf("Enter a positive number: ");
        scanf("%d", &n);
        printf("Fibonacci Series: %d,%d,", t1, t2);
        nextTerm = t1 + t2;
        while (nextTerm <= n) {
                printf("%d,", nextTerm);
                t1 = t2;
                t2 = nextTerm;
                nextTerm = t1 + t2;
        }
        printf("\n");
}
```

## Execution Results - All test cases have succeeded!

| Test Case - 1 |
|---|
| **User Output** |
| Enter a positive number: |
| **100** |
| Fibonacci Series: 0,1,1,2,3,5,8,13,21,34,55,89, |

| Test Case - 2 |
|---|
| **User Output** |
| Enter a positive number: |
| **398** |
| Fibonacci Series: 0,1,1,2,3,5,8,13,21,34,55,89,144,233,377, |

| S.No: 50 | Exp. Name: ***Write a program to check whether the given number is prime or not (number should be passed as an argument)*** | Date: 2023-11-10 |
|---|---|---|









**Aim:**

**Write a C program to check whether the given number is PRIME or not (number should be passed as an argument)**

**Source Code:**

a.c

```
#include <stdio.h>
#include <conio.h>
void main()
{
        int num,res=0;
        printf("ENTER A NUMBER: ");
        scanf("%d",&num);
        res=prime(num);
        if(res==0)
                printf("%d IS A PRIME NUMBER",num);
        else
                printf("%d IS NOT A PRIME NUMBER",num);
}
int prime(int n)
{
        int i;
        for(i=2;i<=n/2;i++)
        {
                if(n%i!=0)
                        continue;
                else
                        return 1;
        }
        return 0;
}
```

## Execution Results - All test cases have succeeded!

| Test Case - 1 |
|---|
| **User Output** |
| ENTER A NUMBER: |
| **2** |
| 2 IS A PRIME NUMBER |

| Test Case - 2 |
|---|
| **User Output** |
| ENTER A NUMBER: |
| **4** |

```
4 IS NOT A PRIME NUMBER
```

| S.No: 51 | Exp. Name: ***Program to count number of digits.*** | Date: 2023-11-10 |
|---|---|---|

**Aim:**

**Write a C program to count the number of digits in a number.**

**Source Code:**

digits.c

```
#include <stdio.h>

int main() {
    long long num;
    int count = 0;
    scanf("%lld", &num);

    while (num != 0) {
        num /= 10;
        count++;
    }

    printf("Number of digits: %d\n", count);

    return 0;
}
```

## Execution Results - All test cases have succeeded!

| Test Case - 1 |
|---|
| **User Output** |
| **12345** |
| Number of digits: 5 |

| Test Case - 2 |
|---|
| **User Output** |
| **987654321** |
| Number of digits: 9 |

| S.No: 52 | Exp. Name: ***Print the multiplication table of a number*** | Date: 2023-11-10 |
|---|---|---|

**Aim:**

**Write a Cprogram toprint the multiplication table of a number.**

**Source Code:**

multiplicationTable.c

```
// Type Content here...
#include <stdio.h>

int main() {
    int num;

    printf("Enter a number: ");
    scanf("%d", &num);

      for (int i = 1; i <= 10; i++) {
        printf("%d * %d = %d\n", num, i, num * i);
    }

    return 0;
}
```

## Execution Results - All test cases have succeeded!

| Test Case - 1 |
|---|
| **User Output** |
| Enter a number: |
| **5** |
| 5 * 1 = 5 |
| 5 * 2 = 10 |
| 5 * 3 = 15 |
| 5 * 4 = 20 |
| 5 * 5 = 25 |
| 5 * 6 = 30 |
| 5 * 7 = 35 |
| 5 * 8 = 40 |
| 5 * 9 = 45 |
| 5 * 10 = 50 |

| Test Case - 2 |
|---|
| **User Output** |
| Enter a number: |
| **9** |
| 9 * 1 = 9 |
| 9 * 2 = 18 |
| 9 * 3 = 27 |

```
9 * 4 = 36
9 * 5 = 45
9 * 6 = 54
9 * 7 = 63
9 * 8 = 72
9 * 9 = 81
9 * 10 = 90
```

| S.No: 53 | Exp. Name: ***Program to find the largest number.*** | Date: 2023-11-10 |
|---|---|---|

**Aim:**

**Write a C program to find the largest number among a series of numbers.**

**Source Code:**

largest.c

```
#include <stdio.h>

int main() {
    double num, largest = -999999; // Initialize 'largest' to a very small value

    printf("Enter a series of numbers (enter 0 to exit):\n");

    while (1) {
        scanf("%lf", &num);

        if (num == 0) {
            break; // Exit the loop when 0 is entered
        }

        if (num > largest) {
            largest = num;
        }
    }

    if (largest == -999999) {
        printf("No numbers entered.\n");
    } else {
        printf("The largest number is %.2lf\n", largest);
    }

    return 0;
}
```

## Execution Results - All test cases have succeeded!

| Test Case - 1 |
|---|
| **User Output** |
| Enter a series of numbers (enter 0 to exit): |
| **4** |
| **10** |
| **7** |
| **0** |
| The largest number is 10.00 |

| Test Case - 2 |
|---|
| **User Output** |

| Enter a series of numbers (enter 0 to exit): |
| --- |
| **45** |
| **67** |
| **34** |
| **87** |
| **988** |
| **3445** |
| **678** |
| **4445** |
| **0** |
| The largest number is 4445.00 |

| S.No: 54 | Exp. Name: ***Program to convert decimal number to its equivalent binary number*** | Date: 2023-11-10 |
|---|---|---|









**Aim:**

**Write a C program to convert a decimal number to its equivalent binary number**

**Source Code:**

decibinary.c

```
#include <stdio.h>

int main() {
    int decimalNum, binaryNum[32], i = 0;

    printf("Enter the number to convert: ");
    scanf("%d", &decimalNum);

    if (decimalNum == 0) {
        printf("Binary of given number is: 0\n");
    } else {
        while (decimalNum > 0) {
            binaryNum[i] = decimalNum % 2;
            decimalNum /= 2;
            i++;
        }

        printf("Binary of given number is: ");
        for (int j = i - 1; j >= 0; j--) {
            printf("%d", binaryNum[j]);
        }
        printf("\n");
    }

    return 0;
}
```

## Execution Results - All test cases have succeeded!

| Test Case - 1 |
|---|
| **User Output** |
| Enter the number to convert: |
| **7** |
| Binary of given number is: 111 |

| Test Case - 2 |
|---|
| **User Output** |
| Enter the number to convert: |
| **48** |
| Binary of given number is: 110000 |

| S.No: 55 | Exp. Name: ***Add numbers until user input is zero using do while*** | Date: 2023-11-10 |
|---|---|---|





**Aim:**

**Write a C program to add numbers until the user enter zero using the do-while loop**

**Source Code:**

addnum.c

```
// write your code here...
#include <stdio.h>
void main() {
        double number, sum = 0;
        printf("Enter zero if u want to exit\n");
        do {
                printf("Enter number: ");
                scanf("%lf", &number);
                sum +=  number;
        }while(number  != 0.0);
        printf("Sum of the numbers entered = %.2lf\n", sum);
}
```

## Execution Results - All test cases have succeeded!

| Test Case - 1 |
|---|
| **User Output** |
| Enter zero if u want to exit |
| Enter number: |
| **26** |
| Enter number: |
| **59** |
| Enter number: |
| **12** |
| Enter number: |
| **332** |
| Enter number: |
| **0** |
| Sum of the numbers entered = 429.00 |

| Test Case - 2 |
|---|
| **User Output** |
| Enter zero if u want to exit |
| Enter number: |
| **594.24** |
| Enter number: |
| **1247.365** |
| Enter number: |

```
784.14
Enter number:
3177.128
Enter number:
0
Sum of the numbers entered = 5802.87
```

| S.No: 56 | Exp. Name: ***Print positive integers between two numbers*** | Date: 2023-11-10 |
|---|---|---|

**Aim:**

**Write a C program that prints positive integers between two numbers using a do-while loop**

**Source Code:**

sample.c

```
#include <stdio.h>
void main() {
        int a, b;
        printf("Enter first integer: ");
        scanf("%d", &a);
        printf("Enter second positive integer greater than first one: ");
        scanf("%d", &b);
        printf("Values are: ");
        do {
                printf("%d ", a);
                a = a + 1;
        } while(a <= b);
        printf("\n");
}
```

## Execution Results - All test cases have succeeded!

| Test Case - 1 |
|---|
| **User Output** |
| Enter first integer: |
| **10** |
| Enter second positive integer greater than first one: |
| **19** |
| Values are: 10 11 12 13 14 15 16 17 18 19 |

| Test Case - 2 |
|---|
| **User Output** |
| Enter first integer: |
| **1** |
| Enter second positive integer greater than first one: |
| **23** |
| Values are: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 |

| S.No: 57 | Exp. Name: ***Print numbers from 1 to n using for loop*** | Date: 2023-11-10 |
|---|---|---|

**Aim:**

**Write a C program that prints numbers from 1 to n using for loop**

**Source Code:**

for.c

```
#include <stdio.h>

int main() {
    int n;

    printf("Enter a number: ");
    scanf("%d", &n);

    printf("Values are: ");
    for (int i = 1; i <= n; i++) {
        printf("%d ", i);
    }
    printf("\n");

    return 0;
}
```

## Execution Results - All test cases have succeeded!

| Test Case - 1 |
|---|
| **User Output** |
| Enter a number: |
| **5** |
| Values are: 1 2 3 4 5 |

| Test Case - 2 |
|---|
| **User Output** |
| Enter a number: |
| **50** |
| Values are: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 |

| S.No: 58 | Exp. Name: ***Print sum of n natural numbers*** | Date: 2023-11-10 |
|---|---|---|

**Aim:**

**Write a C program to print the sum of n natural numbers**

**Source Code:**

sumnatural.c

```
#include <stdio.h>

int main() {
    int n, sum = 0;

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

    // Check if n is positive
    if (n <= 0) {
        printf("Please enter a positive integer.\n");
    } else {
        // Calculate the sum of the first 'n' natural numbers
        sum = (n * (n + 1)) / 2;

        printf("Sum of the numbers = %d\n", sum);
    }

    return 0;
}
```

## Execution Results - All test cases have succeeded!

| Test Case - 1 |
|---|
| **User Output** |
| Enter a positive integer: |
| **10** |
| Sum of the numbers = 55 |

| Test Case - 2 |
|---|
| **User Output** |
| Enter a positive integer: |
| **57** |
| Sum of the numbers = 1653 |

| S.No: 59 | Exp. Name: ***Print pascal triangle*** | Date: 2023-11-10 |
|---|---|---|

**Aim:**

**Write a C program to print the pascal triangle of numbers**

**Source Code:**

pascal.c

```
#include <stdio.h>
void main() {
        int rows, coef = 1, space, i, j;
        printf("Enter the number of rows: ");
        scanf("%d", &rows);
        for (i = 0; i < rows; i++) {
                for (space = 1; space <= rows-i; space++) {
                        printf(" ");
                }
                for (j = 0; j <= i; j++) {
                        if (j == 0 || i == 0) {
                                coef = 1;
                        }
                        else {
                                coef = coef * (i - j + 1) / j;
                        }
                        printf("%4d", coef);
                }
                printf("\n");
        }
}
```

## Execution Results - All test cases have succeeded!

| Test Case - 1 |
|---|
| **User Output** |
| Enter the number of rows: |
| **3** |
| 1 |
| 1 1 |
| 1 2 1 |

| Test Case - 2 |
|---|
| **User Output** |
| Enter the number of rows: |
| **9** |
| 1 |
| 1 1 |
| 1 2 1 |
| 1 3 3 1 |

```
          1   4   6   4   1
        1   5  10  10   5   1
      1   6  15  20  15   6   1
    1   7  21  35  35  21   7   1
  1   8  28  56  70  56  28   8   1
```

| S.No: 60 | Exp. Name: ***Program to calculate sum of geometric series*** | Date: 2023-11-10 |
|---|---|---|

**Aim:**

**Program to find the sum of geometric series**

**S = a+ar+ar$^2$+ar$^3$+..... +ar$^{n-1}$**

**Source Code:**

geometric.c

```
#include <stdio.h>
float sumgeometric(float a, float r, int n) {
        float sum = 0;
        for (int i = 0; i < n; i++) {
                sum = sum + a;
                a = a * r;
        }
        return sum;
}
void main() {
        int a, n;
        float r;
        printf("Enter the first term a: ");
        scanf("%d", &a);
        printf("Enter the common ratio r: ");
        scanf("%f", &r);
        printf("Enter no.of terms: ");
        scanf("%d", &n);
        printf("Sum of the geometric series: %0.2f\n", sumgeometric(a, r, n));
}
```

## Execution Results - All test cases have succeeded!

| Test Case - 1 |
|---|
| **User Output** |
| Enter the first term a: |
| **1** |
| Enter the common ratio r: |
| **0.5** |
| Enter no.of terms: |
| **5** |
| Sum of the geometric series: 1.94 |

| Test Case - 2 |
|---|
| **User Output** |
| Enter the first term a: |
| **5** |
| Enter the common ratio r: |

| 1 |
| --- |
| Enter no.of terms: |
| 10 |
| Sum of the geometric series: 50.00 |

| S.No: 61 | Exp. Name: ***Program to calculate sin(x).*** | Date: 2023-11-10 |
|---|---|---|









**Aim:**

**Write a C program to calculate the** $\sin\ (x)$ **using sine service expansion upto required number of terms and print the value of** $\sin\ (x)$

$\sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \ldots$

**Source Code:**

sinx.c

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

int main() {
    double x, radians, sinX = 0;
    int n;

    printf("Enter x value: ");
    scanf("%lf", &x);

    printf("Enter the number of terms: ");
    scanf("%d", &n);

    radians = x * M_PI / 180.0;  // Convert degrees to radians

    for (int i = 0; i < n; i++) {
        double term = pow(-1, i) * pow(radians, 2 * i + 1) / tgamma(2 * i + 2);
        sinX += term;
    }

    printf("sin(%.6lf degrees) = %.6lf\n", x, sinX);

    return 0;
}
```

## Execution Results - All test cases have succeeded!

| Test Case - 1 |
|---|
| **User Output** |
| Enter x value: |
| **45** |
| Enter the number of terms: |
| **4** |
| sin(45.000000 degrees) = 0.707106 |

| Test Case - 2 |
|---|
| **User Output** |
| Enter x value: |

| 34 |
| --- |
| Enter the number of terms: |
| 3 |
| sin(34.000000 degrees) = 0.559198 |

| S.No: 62 | Exp. Name: ***Program to print the sum of cos series*** | Date: 2023-11-10 |
|---|---|---|

**Aim:**

**Write a C program to print the sum of Cos series.**

$\cos x = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \dots$

**Source Code:**

cosSum.c

```
#include <stdio.h>
const double PI = 3.142;
//will return the sum of cos(x)
double series_sum(double x, int n) {
    x = x * (PI / 180.0);
    double result = 1;
    double s = 1, fact = 1, pow = 1;
    for (int i = 1; i < 5; i++) {
        s = s * -1;
        fact = fact * (2 * i - 1) * (2 * i);
        pow = pow * x * x;
        result = result + s * pow / fact;
    }
    return result;
}
//main function
int main() {
    double x;
    int n;
    scanf("%lf", &x);
    scanf("%d", &n);
    printf("%lf\n", series_sum(x, n));
    return 0;
}
```

## Execution Results - All test cases have succeeded!

| Test Case - 1 |
|---|
| **User Output** |
| **10** |
| **3** |
| 0.984804 |

| Test Case - 2 |
|---|
| **User Output** |
| **13** |
| **8** |
| 0.974363 |

| S.No: 63 | Exp. Name: ***Read n integer values into an array and display them*** | Date: 2023-11-11 |
|---|---|---|









**Aim:**

**Write a C program to read n integer values into an array and display them horizontally & vertically.**

**Input format:**
**The first line of input should be the size of the array**
**The second line of input should be the integers separated by spaces**
**Output format:**
**The output should be the array of integers in a horizontal & vertical manner.**

**Source Code:**

displayArray.c

```
#include<stdio.h>
int main()
{
        int size;
        scanf("%d",&size);
        int arr[size];
        printf("",size);
        for (int i=0;i<size;i++) {
                scanf("%d",&arr[i]);
        }
        printf("Array in Horizontal manner: ");
        for(int i=0;i<size;i++){
                printf("%d ",arr[i]);
        }
        printf("\n");
        printf("Array in Vertical manner: \n");
        for(int i=0;i<size;i++){
                printf("%d\n",arr[i]);
        }
}
```

## Execution Results - All test cases have succeeded!

| Test Case - 1 |
|---|
| **User Output** |
| **5** |
| **1 2 3 4 5** |
| Array in Horizontal manner: 1 2 3 4 5 |
| Array in Vertical manner: |
| 1 |
| 2 |
| 3 |
| 4 |
| 5 |

| Test Case - 2 |
| --- |
| **User Output** |
| **7** |
| **11 12 13 14 15 16 17** |
| Array in Horizontal manner: 11 12 13 14 15 16 17 |
| Array in Vertical manner: |
| 11 |
| 12 |
| 13 |
| 14 |
| 15 |
| 16 |
| 17 |

| S.No: 64 | Exp. Name: ***Write a C program to find minimum value and maximum value from given elements in an array.*** | Date: 2023-11-11 |
|---|---|---|

**Aim:**

**Write a C program to find minimum value and maximum value from given elements in an array.**

**Source Code:**

MinMax.c

```
#include <stdio.h>

int main() {
    int n, i;
    int min, max;

    printf("Enter array size: ");
    scanf("%d", &n);

    int a[n];

    printf("Enter array elements: ");
    for (i = 0; i < n; i++) {
        scanf("%d", &a[i]);
    }

    min = max = a[0];

    for (i = 1; i < n; i++) {
        if (a[i] < min) {
            min = a[i];
        }
        if (a[i] > max) {
            max = a[i];
        }
    }

    printf("Minimum element: %d\n", min);
    printf("Maximum element: %d\n", max);

    return 0;
}
```

Execution Results - All test cases have succeeded!

| Test Case - 1 |
|---|
| **User Output** |
| Enter array size: |
| **5** |
| Enter array elements: |
| **8 4 36 95 21** |

| Minimum element: 4 |
| --- |
| Maximum element: 95 |

| Test Case - 2 |
| --- |
| **User Output** |
| Enter array size: |
| **10** |
| Enter array elements: |
| **59 36 147 564 879 214 635 45 95 10** |
| Minimum element: 10 |
| Maximum element: 879 |

| S.No: 65 | Exp. Name: ***C Program to implement searching operation on array using Binary Search.*** | **Date: 2023-11-11** |
|---|---|---|

**Aim:**

**C Program to implement searching operation on array using Binary Search.**

**Source Code:**

`BinarySearch.c`

```c
#include <stdio.h>
void read();
void sort();
void binarysearch();
int arr[100], key, n, i, j, low, high, mid;
void main() {
  read();
  sort();

  binarysearch();
}
void read() {
  printf("Enter value of n : ");
  scanf("%d", &n);
  for (i = 0; i < n; i++) {
    printf("Enter element for a[%d] : ", i);
    scanf("%d", &arr[i]);
  }
  printf("Enter key element : ");
  scanf("%d", &key);
}
void sort() {
  for (i = 0; i < n - 1; i++) {
    for (j = 0; j < n - 1 - i; j++) {
      if (arr[j] > arr[j + 1]) {
        int temp;
        temp = arr[j];
        arr[j] = arr[j + 1];
        arr[j + 1] = temp;
      }
    }
  }
}
void binarysearch() {
  printf("After sorting the elements in the array are\n");
  for (i = 0; i < n; i++) {
    printf("Value of a[%d] = ", i);
    printf("%d\n", arr[i]);
  }
  low = 0;
  high = n - 1;
  while (low <= high) {
    mid = (low + high) / 2;
    if (key == arr[mid]) {
      printf("The key element %d is found at the position %d\n", key, mid);
      break;
    } else if (key < arr[mid])
      high = mid - 1;
    else if (key > arr[mid])
      low = mid + 1;
  }
  if (low > high) printf("The Key element %d is not found in the array\n", key);
}
```

## Execution Results - All test cases have succeeded!

| Test Case - 1 |
|---|
| **User Output** |
| Enter value of n : |
| **5** |
| Enter element for a[0] : |
| **4** |
| Enter element for a[1] : |
| **8** |
| Enter element for a[2] : |
| **6** |
| Enter element for a[3] : |
| **2** |
| Enter element for a[4] : |
| **1** |
| Enter key element : |
| **8** |
| After sorting the elements in the array are |
| Value of a[0] = 1 |
| Value of a[1] = 2 |
| Value of a[2] = 4 |
| Value of a[3] = 6 |
| Value of a[4] = 8 |
| The key element 8 is found at the position 4 |

| Test Case - 2 |
|---|
| **User Output** |
| Enter value of n : |
| **7** |
| Enter element for a[0] : |
| **56** |
| Enter element for a[1] : |
| **89** |
| Enter element for a[2] : |
| **63** |
| Enter element for a[3] : |
| **215** |
| Enter element for a[4] : |
| **325** |
| Enter element for a[5] : |
| **156** |
| Enter element for a[6] : |
| **256** |
| Enter key element : |
| **458** |
| After sorting the elements in the array are |

```
Value of a[0] = 56
Value of a[1] = 63
Value of a[2] = 89
Value of a[3] = 156
Value of a[4] = 215
Value of a[5] = 256
Value of a[6] = 325
The Key element 458 is not found in the array
```

| S.No: 66 | Exp. Name: ***Program - Addition of two matrices*** | Date: 2023-11-11 |
|---|---|---|

**Aim:**

**The below sample code finds the `addition` of two matrices.**

**In the `main()` function read a two two-dimensional array of elements and then find the `addition` of two matrices.**

**The `logic` is**

- **First checks the row sizes and column sizes of two two-dimensional arrays are equal or not.**
- **If the sizes are not equal then print "Addition is not possible" and stop the process.**
- **If the sizes are equal then use two for loops to add each corresponding elements of two matrices and finally print the result.**

**Fill in the missing code so that it produces the desired output.**

**Source Code:**

AdditionOfMatrices1.c

```
#include<stdio.h>
void main() {
        int i, j, m, n, p, q;
        int a[5][5], b[5][5], c[5][5];
        printf("Enter the row & column sizes of matrix-1 : ");
        scanf("%d %d", &m, &n);
        printf("Enter matrix-1 %d elements : ", m*n);
        for ( i=0;i<m;i++) { // Complete the code in for
                for (j=0;j<n;j++ ) { // Complete the code in for
                        scanf("%d", &a[i][j]);
                }
        }
        printf("Enter the row & column sizes of matrix-2 : ");
        scanf("%d %d", &p, &q);
        printf("Enter matrix-2 %d elements : ", p*q);
        for (i=0;i<p;i++ ) { // Complete the code in for
                for (j=0;j<q;j++ ) { // Complete the code in for
                        scanf("%d", &b[i][j]);
                }
        }
        printf("The given matrix-1 is\n");
        for ( i=0;i<m;i++)
        { // Complete the code in for
                for (j=0;j<n;j++ )
                { // Complete the code in for
                        printf("%d ", a[i][j]);
                }
                printf("\n");
        }
        // Write the code to display Matrix-1 elements
        printf("The given matrix-2 is\n");
        for ( i=0;i<p;i++)
        { // Complete the code in for
                for (j=0;j<q;j++ )
                { // Complete the code in for
                        printf("%d ", b[i][j]);
                }
                        printf("\n");
        }
        // Write the code to display Matrix-2 elements
        if ( m==p && n==q)
        { // Write the condition part
                for (i=0;i<m;i++ )
                { // Complete the code in for
                        for (j=0;j<n;j++ )
                        { // Complete the code in for
                                c[i][j] = a[i][j]+b[i][j]; // Complete the statement
                        }
                }
                printf("Addition of two matrices is\n");
                for (i=0;i<p;i++)
                { // Complete the code in for
                        for (j=0;j<q;j++ )
                        { // Complete the code in for
                                printf("%d ", c[i][j]);
```

```
                }
                // Write the code to display resultant matrix elements
        }
        else
        {
                printf("Addition is not possible\n");
        }
}
```

## Execution Results - All test cases have succeeded!

| Test Case - 1 |
| --- |
| **User Output** |
| Enter the row & column sizes of matrix-1 : |
| **2 2** |
| Enter matrix-1 4 elements : |
| **1 2 3 4** |
| Enter the row & column sizes of matrix-2 : |
| **2 2** |
| Enter matrix-2 4 elements : |
| **4 5 6 7** |
| The given matrix-1 is |
| 1 2 |
| 3 4 |
| The given matrix-2 is |
| 4 5 |
| 6 7 |
| Addition of two matrices is |
| 5 7 |
| 9 11 |

| S.No: 67 | Exp. Name: ***Write a C program to find the Transpose of a given matrix*** | Date: 2023-11-11 |
|---|---|---|









**Aim:**

**Write a program to find the `transpose` of a given matrix.**

**[Hint: A transpose matrix writes the rows of a given matrix as the columns of the matrix.]**

**At the time of execution, the program should print the message on the console as:**

```
Enter the order of matrix :
```

**For example, if the user gives the input as:**

```
Enter the order of matrix : 2 3
```

**Next, the program should print the message on the console as:**

```
Enter 6 elements :
```

**if the user gives the input as:**

```
Enter 6 elements : 1 2 3 4 5 6
```

**then the program should print the result as:**

```
The given matrix is
1 2 3
4 5 6
Transpose of the given matrix is
1 4
2 5
3 6
```

**Note: Do use the printf() function with a newline character (`\n`).**

**Source Code:**

Program503.c

```c
#include <stdio.h>

int main() {
    int matrix[10][10], transpose[10][10];
    int row, col, i, j;

    // Input for matrix
    printf("Enter the order of matrix : ");
    scanf("%d %d", &row, &col);
    printf("Enter %d elements : ", row * col);
    for (i = 0; i < row; ++i) {
        for (j = 0; j < col; ++j) {
            scanf("%d", &matrix[i][j]);
        }
    }

    // Printing the given matrix
    printf("The given matrix is\n");
    for (i = 0; i < row; ++i) {
        for (j = 0; j < col; ++j) {
            printf("%d ", matrix[i][j]);
        }
        printf("\n");
    }

    // Finding transpose of the matrix
    for (i = 0; i < col; ++i) {
        for (j = 0; j < row; ++j) {
            transpose[i][j] = matrix[j][i];
        }
    }

    // Printing the transpose matrix
    printf("Transpose of the given matrix is\n");
    for (i = 0; i < col; ++i) {
        for (j = 0; j < row; ++j) {
            printf("%d ", transpose[i][j]);
        }
        printf("\n");
    }

    return 0;
}
```

## Execution Results - All test cases have succeeded!

| Test Case - 1 |
| --- |
| **User Output** |
| Enter the order of matrix : |
| **3 2** |
| Enter 6 elements : |
| **11 22 44 66 77 99** |

| The given matrix is |
|---|
| 11 22 |
| 44 66 |
| 77 99 |
| Transpose of the given matrix is |
| 11 44 77 |
| 22 66 99 |

| Test Case - 2 |
|---|
| **User Output** |
| Enter the order of matrix : |
| **2 2** |
| Enter 4 elements : |
| **25 45 65 85** |
| The given matrix is |
| 25 45 |
| 65 85 |
| Transpose of the given matrix is |
| 25 65 |
| 45 85 |

| Test Case - 3 |
|---|
| **User Output** |
| Enter the order of matrix : |
| **2 4** |
| Enter 8 elements : |
| **1 2 3 4 5 6 7 8** |
| The given matrix is |
| 1 2 3 4 |
| 5 6 7 8 |
| Transpose of the given matrix is |
| 1 5 |
| 2 6 |
| 3 7 |
| 4 8 |

| S.No: 68 | Exp. Name: ***Write a C program to find the Multiplication of Two matrices by checking compatibility*** | Date: 2023-11-11 |
|---|---|---|









**Aim:**

**Develop, implement and execute a C program that reads two matrices A (m x n ) and B (p x q) and compute the `product` of A and B.**

**Read matrix A, matrix B and print both the input matrices and resultant matrix with suitable headings and in matrix format.**

**Program must check the compatibility of orders of the matrices for multiplication. Report appropriate message in case of incompatibility.**

**At the time of execution, the program should print the message on the console as:**

```
Enter the row & column sizes of matrix-1 :
```

**For example, if the user gives the input as:**

```
Enter the row & column sizes of matrix-1 : 3 2
```

**Next, the program should print the message on the console as:**

```
Enter the row & column sizes of matrix-2 :
```

**if the user gives the input as:**

```
Enter the row & column sizes of matrix-2 : 2 3
```

**Next, the program should print the message on the console as:**

```
Enter matrix-1 6 elements :
```

**if the user gives the input as:**

```
Enter matrix-1 6 elements : 1 2 3 4 5 6
```

**Next, the program should print the message on the console as:**

```
Enter matrix-2 6 elements :
```

**if the user gives the input as:**

```
Enter matrix-2 6 elements : 4 5 6 7 8 9
```

**then the program should print the result as:**

```
The given matrix-1 is
1 2 3
4 5 6
The given matrix-2 is
4 5
6 7
8 9
Multiplication of two matrices is
40 46
94 109
```

**If the input is given as** `3 2` **and** `3 2` **then the program should print the result as:**

```
Multiplication is not possible
```

**Source Code:**

`Lab8.c`

```c
#include <stdio.h>

int main() {
    int m, n, p, q, i, j, k;

    // Input sizes of matrices A and B
    printf("Enter the row & column sizes of matrix-1 : ");
    scanf("%d %d", &m, &n);
    printf("Enter the row & column sizes of matrix-2 : ");
    scanf("%d %d", &p, &q);

    // Check for compatibility of matrices for multiplication
    if (n != p) {
        printf("Multiplication is not possible\n");
        return 0;
    }

    int A[m][n], B[p][q], result[m][q];

    // Input elements of matrix A
    printf("Enter matrix-1 %d elements : ", m * n);
    for(i = 0; i < m; i++) {
        for(j = 0; j < n; j++) {
            scanf("%d", &A[i][j]);
        }
    }

    // Input elements of matrix B
    printf("Enter matrix-2 %d elements : ", p * q);
    for(i = 0; i < p; i++) {
        for(j = 0; j < q; j++) {
            scanf("%d", &B[i][j]);
        }
    }

    // Multiply matrices A and B
    for(i = 0; i < m; i++) {
        for(j = 0; j < q; j++) {
            result[i][j] = 0;
            for(k = 0; k < n; k++) {
                result[i][j] += A[i][k] * B[k][j];
            }
        }
    }
        // Print matrices A, B, and the result
    printf("The given matrix-1 is\n");
    for(i = 0; i < m; i++) {
        for(j = 0; j < n; j++) {
            printf("%d ", A[i][j]);
        }
        printf("\n");
    }

    printf("The given matrix-2 is\n");
    for(i = 0; i < p; i++) {
```

```
        }
        printf("\n");
    }
    printf("Multiplication of two matrices is\n");
    for(i = 0; i < m; i++) {
        for(j = 0; j < q; j++) {
            printf("%d ", result[i][j]);
        }
        printf("\n");
    }

    return 0;
}
```



## Execution Results - All test cases have succeeded!

| Test Case - 1 |
|---|
| **User Output** |
| Enter the row & column sizes of matrix-1 : |
| **3 2** |
| Enter the row & column sizes of matrix-2 : |
| **2 3** |
| Enter matrix-1 6 elements : |
| **1 2 3 4 5 6** |
| Enter matrix-2 6 elements : |
| **1 2 3 4 5 6** |
| The given matrix-1 is |
| 1 2 |
| 3 4 |
| 5 6 |
| The given matrix-2 is |
| 1 2 3 |
| 4 5 6 |
| Multiplication of two matrices is |
| 9 12 15 |
| 19 26 33 |
| 29 40 51 |

| S.No: 69 | Exp. Name: ***Write a C Program to find the sum and average marks of a student using arrays.*** | Date: 2023-11-11 |
|---|---|---|

**Aim:**

**Write a C Program to find the sum and average marks of a student using arrays.**

**Source Code:**

SumAvg.c

```
#include <stdio.h>

int main() {
    int numSubjects, i;
    float sum = 0;

    // Input number of subjects
    printf("Enter number of subjects of a student: ");
    scanf("%d", &numSubjects);

    int marks[numSubjects];

    // Input marks for each subject
    printf("Enter marks: ");
    for (i = 0; i < numSubjects; i++) {
        scanf("%d", &marks[i]);
        sum += marks[i]; // Calculate sum of marks
    }

    // Calculate average marks
    float average = sum / numSubjects;

    // Display sum and average marks
    printf("Sum of marks: %.2f\n", sum);
    printf("Average of marks: %.2f\n", average);

    return 0;
}
```

## Execution Results - All test cases have succeeded!

| Test Case - 1 |
|---|
| **User Output** |
| Enter number of subjects of a student: |
| **4** |
| Enter marks: |
| **5 2 3 1** |
| Sum of marks: 11.00 |
| Average of marks: 2.75 |

| Test Case - 2 |
|---|

| User Output |
| --- |
| Enter number of subjects of a student: |
| **9** |
| Enter marks: |
| **25 59 14 36 74 52 89 15 67** |
| Sum of marks: 431.00 |
| Average of marks: 47.89 |

| S.No: 70 | Exp. Name: ***Write a C program to Search a Key element using Linear search Technique*** | Date: 2023-11-11 |
|---|---|---|









**Aim:**

**Write a program to search a key element with in the given array of elements using `linear search` process.**

**At the time of execution, the program should print the message on the console as:**

```
Enter value of n :
```

**For example, if the user gives the input as:**

```
Enter value of n : 3
```

**Next, the program should print the messages one by one on the console as:**

```
Enter element for a[0] :
Enter element for a[1] :
Enter element for a[2] :
```

**if the user gives the input as:**

```
Enter element for a[0] : 89
Enter element for a[1] : 33
Enter element for a[2] : 56
```

**Next, the program should print the message on the console as:**

```
Enter key element :
```

**if the user gives the input as:**

```
Enter key element : 56
```

**then the program should print the result as:**

```
The key element 56 is found at the position 2
```

**Similarly if the key element is given as 25 for the above one dimensional array elements then the program should print the output as "The key element 25 is not found in the array".**

**Fill in the missing code so that it produces the desired result.**

**Source Code:**

`LinearSearchDemo3.c`

```
#include<stdio.h>
void main() {
        int a[20], i, n, key, flag = 0, pos;
        printf("Enter value of n : ");
        scanf("%d", &n);
        // Write code to read array elements
         for (int i = 0; i < n; i++) {
        printf("Enter element for a[%d] : ", i);
        scanf("%d", &a[i]);
    }

        printf("Enter key element : ");
        scanf("%d", &key);
        // Write code for linear search process
         for (int i = 0; i < n; i++) {
        if (a[i] == key) {
            printf("The key element %d is found at the position %d\n", key, i);
            flag = 1;
            break;
        }
    }


        if (flag==0) { //Write the condition part

                printf("The key element %d is not found in the array\n",key ); //Complete
the statement
        }
}
```

## Execution Results - All test cases have succeeded!

| Test Case - 1 |
|---|
| **User Output** |
| Enter value of n : |
| **4** |
| Enter element for a[0] : |
| **1** |
| Enter element for a[1] : |
| **22** |
| Enter element for a[2] : |
| **33** |
| Enter element for a[3] : |
| **44** |
| Enter key element : |
| **22** |
| The key element 22 is found at the position 1 |

| Test Case - 2 |
| --- |
| **User Output** |
| Enter value of n : |
| **6** |
| Enter element for a[0] : |
| **2** |
| Enter element for a[1] : |
| **4** |
| Enter element for a[2] : |
| **8** |
| Enter element for a[3] : |
| **1** |
| Enter element for a[4] : |
| **3** |
| Enter element for a[5] : |
| **5** |
| Enter key element : |
| **6** |
| The key element 6 is not found in the array |

| S.No: 71 | Exp. Name: ***Write a C program to display the elements of an Array in reverse order*** | Date: 2023-11-11 |
|---|---|---|









**Aim:**

**Write a program to print the given integer elements of an array (with max size 10) in `reverse order`.**

**At the time of execution, the program should print the message on the console as:**

```
Enter size of the array :
```

**For example, if the user gives the input as:**

```
Enter size of the array : 3
```

**Next, the program should print the message on the console as:**

```
Enter array elements :
```

**If the user gives the input as:**

```
Enter array elements : 10 20 30
```

**then the program should print the result as:**

```
Array elements in reverse order : 30 20 10
```

**[Hint: First read an integers from standard input into the array and then use a loop to iterate on that array in the reverse order (meaning starting from the last element till the first) to print the elements.]**

**Note: Do use the printf() function without a newline character (`\n`).**

**Source Code:**

`ArrayAccessDemo1.c`

```
/*#include <stdio.h>
void main() {
        int arr[10], i, n;
        printf("Enter size of the array : ");
        scanf("%d",  );
        printf("Enter array elements : ");
        // Fill the missing code


        printf("Array elements in reverse order : ");
        // Fill the missing code


}       */
#include <stdio.h>

int main() {
    int size, i;

    // Input size of the array
    printf("Enter size of the array : ");
    scanf("%d", &size);

    // Ensure size is within the allowed limit
    if (size <= 0 || size > 10) {
        printf("Invalid size of the array.\n");
        return 0;
    }

    int array[size];

    // Input array elements
    printf("Enter array elements : ");
    for (i = 0; i < size; i++) {
        scanf("%d", &array[i]);
    }

    // Print array elements in reverse order
    printf("Array elements in reverse order : ");
    for (i = size - 1; i >= 0; i--) {
        printf("%d ", array[i]);
    }

    return 0;
}
```

## Execution Results - All test cases have succeeded!

| Test Case - 1 |
| --- |
| **User Output** |
| Enter size of the array : |
| **3** |

| Enter array elements : |
|---|
| **10 20 30** |
| Array elements in reverse order : 30 20 10 |

| Test Case - 2 |
|---|
| **User Output** |
| Enter size of the array : |
| **6** |
| Enter array elements : |
| **11 88 66 22 33 44** |
| Array elements in reverse order : 44 33 22 66 88 11 |

| Test Case - 3 |
|---|
| **User Output** |
| Enter size of the array : |
| **4** |
| Enter array elements : |
| **123 345 675 845** |
| Array elements in reverse order : 845 675 345 123 |

| Test Case - 4 |
|---|
| **User Output** |
| Enter size of the array : |
| **3** |
| Enter array elements : |
| **-23 -45 -99** |
| Array elements in reverse order : -99 -45 -23 |

| S.No: 72 | Exp. Name: ***Write a C Program to calculate the Median for the given values.*** | **Date: 2023-11-11** |
|---|---|---|

**Aim:**

**Write a C Program to calculate the Median for the given values.**

**Source Code:**

`Median.c`

```c
#include <stdio.h>

// function to sort the array in ascending order
void Array_sort(int *array , int n)
{
    // declare some local variables
    int i=0 , j=0 , temp=0;

    for(i=0 ; i<n ; i++)
    {
        for(j=0 ; j<n-1 ; j++)
        {
            if(array[j]>array[j+1])
            {
                temp        = array[j];
                array[j]    = array[j+1];
                array[j+1]  = temp;
            }
        }
    }

    printf("Array After Sorting :");
    for(i=0 ; i<n ; i++)
    {
        printf("\n%d",array[i]);
    }
}

// function to calculate the median of the array
float Find_median(int array[] , int n)
{
    float median=0;

    // if number of elements are even
    if(n%2 == 0)
        median = (array[(n-1)/2] + array[n/2])/2.0;
    // if number of elements are odd
    else
        median = array[n/2];

    return median;
}

int main()
{
    // declare two int arrays
    int array_1[30] = {0};
    // declare some local variables
    int i=0 ,n=0, median=0;

    printf("Enter Array Size: ");
    scanf("%d",&n);

    printf("Enter Array Elements :\n");
    for(i=0 ; i<n ; i++)
```

```
    }

    // Sort the array in ascending order
    Array_sort(array_1 , n);

    // Now pass the sorted array to calculate
    // the median of your array.
    median = Find_median(array_1 , n);

    printf("\nMedian: %d\n",median);
    return 0;
}
```



## Execution Results - All test cases have succeeded!

| Test Case - 1 |
|---|
| **User Output** |
| Enter Array Size: |
| **4** |
| Enter Array Elements : |
| **5** |
| **2** |
| **6** |
| **1** |
| Array After Sorting : |
| 1 |
| 2 |
| 5 |
| 6 |
| Median: 3 |

| Test Case - 2 |
|---|
| **User Output** |
| Enter Array Size: |
| **3** |
| Enter Array Elements : |
| **5** |
| **9** |
| **2** |
| Array After Sorting : |
| 2 |
| 5 |
| 9 |
| Median: 5 |

| S.No: 73 | Exp. Name: ***Write a C program to print a Multiplication Table*** | Date: 2023-11-11 |
|---|---|---|





**Aim:**

**Write a C program to print a Multiplication Table as below :**

MULTIPLICATION TABLE

| | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| 1\| | 1 | 2 | 3 | 4 | 5 |
| 2\| | 2 | 4 | 6 | 8 | 10 |
| 3\| | 3 | 6 | 9 | 12 | 15 |
| 4\| | 4 | 8 | 12 | 16 | 29 |
| 5\| | 5 | 10 | 15 | 20 | 25 |

**Source Code:**

`multiplicationTable.c`

```c
#include<stdio.h>
// #define Row 25
// #define Col 25
int main()
{
    int r,c,Row,Col;

    printf("Enter Number for Multiplication Table :");
    scanf("%d",&Row);

    Col = Row;

    int mul[Row][Col];

    // printf("\t\t\t\t====================\n");
    printf("MULTIPLICATION TABLE\n");
    // printf("---------------------------------");
    printf("     ");
    for(int j=1;j<=Col;j++)
    {

        printf("%4d",j);
    }
    printf("\n-------------------------------------------------\n");
    // printf("=================\n");

    for(int i=0;i<Row;i++)
    {
        r=i+1;
        printf("%2d | ", r);

        for(int j=1;j<=Col;j++)
        {
            c=j;
            mul[i][j]=r*c;
            printf("%4d",mul[i][j]);
        }
        printf("\n");
    }
    return 0;
}
```

## Execution Results - All test cases have succeeded!

| Test Case - 1 |
|---|
| **User Output** |
| Enter Number for Multiplication Table : |
| **3** |
| MULTIPLICATION TABLE |
| 1 2 3 |
| ------------------------------------------------ |
| 1 \| 1 2 3 |

```
 2 |    2   4   6
 3 |    3   6   9
```

**Test Case - 2**

**User Output**

```
Enter Number for Multiplication Table :
10
MULTIPLICATION TABLE
        1   2   3   4   5   6   7   8   9  10
-------------------------------------------------
 1 |    1   2   3   4   5   6   7   8   9  10
 2 |    2   4   6   8  10  12  14  16  18  20
 3 |    3   6   9  12  15  18  21  24  27  30
 4 |    4   8  12  16  20  24  28  32  36  40
 5 |    5  10  15  20  25  30  35  40  45  50
 6 |    6  12  18  24  30  36  42  48  54  60
 7 |    7  14  21  28  35  42  49  56  63  70
 8 |    8  16  24  32  40  48  56  64  72  80
 9 |    9  18  27  36  45  54  63  72  81  90
10 |   10  20  30  40  50  60  70  80  90 100
```

| S.No: 74 | Exp. Name: ***Write a program to check whether the given Matrix is an Identity Matrix or not using Functions*** | Date: 2023-11-11 |
|---|---|---|









**Aim:**

**Write a program to check whether the given matrix is an `identity matrix` or not using functions.**

**At the time of execution, the program should print the message on the console as:**

```
Enter the size of square matrix :
```

**For example, if the user gives the input as:**

```
Enter the size of square matrix : 3
```

**Next, the program should print the message on the console as:**

```
Enter 9 elements :
```

**if the user gives the input as:**

```
Enter 9 elements : 1 0 0 0 1 0 0 0 1
```

**then the program should print the result as:**

```
The given matrix is
1 0 0
0 1 0
0 0 1
It is an identity matrix
```

**Note: Do use the printf() function with a newline character (`\n`).**

**Source Code:**

FunctionsAndArrays5.c

```c
#include <stdio.h>

int isIdentityMatrix(int matrix[], int size) {
    int i, j;
    for (i = 0; i < size; i++) {
        for (j = 0; j < size; j++) {
            if ((i == j && matrix[i * size + j] != 1) || (i != j && matrix[i * size + j] !=
0)) {
                return 0; // If the conditions for an identity matrix are not met, return
false (0)
            }
        }
    }
    return 1; // If all conditions are met, return true (1)
}

void printMatrix(int matrix[], int size) {
    int i, j;
    for (i = 0; i < size; i++) {
        for (j = 0; j < size; j++) {
            printf("%d ", matrix[i * size + j]);
        }
        printf("\n");
    }
}

int main() {
    int size, i;

    // Input size of the square matrix
    printf("Enter the size of square matrix : ");
    scanf("%d", &size);

    // Ensure size is within the allowed limit
    if (size <= 0 || size > 10) {
        printf("Invalid size of the matrix.\n");
        return 0;
    }

    int matrix[size * size];

    // Input matrix elements
    printf("Enter %d elements : ", size * size);
    for (i = 0; i < size * size; i++) {
        scanf("%d", &matrix[i]);
    }

    // Print the given matrix
    printf("The given matrix is\n");
    printMatrix(matrix, size);

    // Check if the matrix is an identity matrix
    if (isIdentityMatrix(matrix, size)) {
        printf("It is an identity matrix\n");
    } else {
```

```
    return 0;
}
```

## Execution Results - All test cases have succeeded!

| Test Case - 1 |
|---|
| **User Output** |
| Enter the size of square matrix : |
| **3** |
| Enter 9 elements : |
| **1 0 0 0 1 0 0 0 1** |
| The given matrix is |
| 1 0 0 |
| 0 1 0 |
| 0 0 1 |
| It is an identity matrix |

| Test Case - 2 |
|---|
| **User Output** |
| Enter the size of square matrix : |
| **2** |
| Enter 4 elements : |
| **1 0 0 1** |
| The given matrix is |
| 1 0 |
| 0 1 |
| It is an identity matrix |

| Test Case - 3 |
|---|
| **User Output** |
| Enter the size of square matrix : |
| **3** |
| Enter 9 elements : |
| **1 0 0 0 1 0 0 0 2** |
| The given matrix is |
| 1 0 0 |
| 0 1 0 |
| 0 0 2 |
| It is not an identity matrix |

| Test Case - 4 |
|---|
| **User Output** |

```
Enter the size of square matrix :
2
Enter 4 elements :
1 2 3 4
The given matrix is
1 2
3 4
It is not an identity matrix
```

| S.No: 75 | Exp. Name: ***Write a C program to find a trace of a matrix.*** | Date: 2023-11-11 |
|---|---|---|

**Aim:**

**Write a C program to find a trace of a matrix.**

**Note :Trace is possible only for a square matrix.**

**Source Code:**

trace.c

```
#include <stdio.h>

int main() {
    int a[5][5],rows, columns, i, j,sum=0;

    // Input number of rows and columns
    printf("Enter number of rows and columns:");
    scanf("%d %d", &rows, &columns);

    // Check if it's a square matrix
    if (rows == columns) {
        printf("Enter Elements in Matrix:-\n",rows,columns);
        for(i=0;i<rows;i++)
        {
  for(j=0;j<columns;j++)
      scanf("%d",&a[i][j]);
  }
       for(i=0;i<rows;i++)
     sum=sum+a[i][i];
  printf("Trace of the matrix = %d",sum);
      }
     else
  printf("Not a square matrix");
   return 0;
  }
```

## Execution Results - All test cases have succeeded!

| Test Case - 1 |
|---|
| **User Output** |
| Enter number of rows and columns: |
| **3** |
| **3** |
| Enter Elements in Matrix:- |
| **1 2 3** |
| **4 5 6** |
| **7 8 9** |
| Trace of the matrix = 15 |

| Test Case - 2 |
| --- |
| **User Output** |
| Enter number of rows and columns: |
| **3** |
| **4** |
| Not a square matrix |

| S.No: 76 | Exp. Name: ***Write a C program to print Upper Triangular Matrix for a given matrix.*** | Date: 2023-11-11 |
|---|---|---|

**Aim:**

**Write a C program to print Upper Triangular Matrix for a given matrix.**

**Source Code:**

upperTriagularMatrix.c

```
/* C Program to find Upper Triangle Matrix */

#include<stdio.h>

int main()
{
        int i, j, rows, columns, a[10][10];

        printf("Enter Number of rows and columns  :");
        scanf("%d %d", &i, &j);

        printf("Enter the Matrix Elements:\n");
        for(rows = 0; rows < i; rows++)
        {
                for(columns = 0;columns < j;columns++)
        {
                scanf("%d", &a[rows][columns]);
        }
        }
     printf("Upper Triagular Matrix:");
        for(rows = 0; rows < i; rows++)
        {
                printf("\n");
                for(columns = 0; columns < j; columns++)
        {
                if(columns >= rows)
                {
                        printf("%d ", a[rows][columns]);
                        }
                        else
                        {
                                printf("0 ");
                        }
                }
        }

        return 0;
}
```

## Execution Results - All test cases have succeeded!

| Test Case - 1 |
|---|
| **User Output** |

| Enter Number of rows and columns : |
|---|
| **3** |
| **3** |
| Enter the Matrix Elements: |
| **12 45 36** |
| **78 69 90** |
| **60 70 30** |
| Upper Triagular Matrix: |
| 12 45 36 |
| 0 69 90 |
| 0 0 30 |

| Test Case - 2 |
|---|
| **User Output** |
| Enter Number of rows and columns : |
| **2** |
| **2** |
| Enter the Matrix Elements: |
| **14 78** |
| **39 46** |
| Upper Triagular Matrix: |
| 14 78 |
| 0 46 |

| S.No: 77 | Exp. Name: ***Write a Program to check whether the given Matrix is Symmetric or not*** | Date: 2023-11-11 |
|---|---|---|









**Aim:**

**Write a C program to find whether a given matrix is a `symmetric matrix` or not.**

**Hint: A symmetric matrix is a square matrix that is equal to its transpose.**

**At the time of execution, the program should print the message on the console as:**

```
Enter the order of matrix :
```

**For example, if the user gives the input as:**

```
Enter the order of matrix : 2 2
```

**Next, the program should print the message on the console as:**

```
Enter 4 elements :
```

**if the user gives the input as:**

```
Enter 4 elements : 4 5 5 4
```

**then the program should print on the console as:**

```
The given matrix is
4 5
5 4
Transpose of the given matrix is
4 5
5 4
The given matrix is symmetric matrix
```

**If the condition is true, then the program should print the result as :**

```
The given matrix is symmetric matrix
```

**Otherwise, the program should print the result as :**

```
The given matrix is not symmetric matrix
```

**Note: Do use the printf() function with a newline character (\n).**

**Source Code:**

SymmetricMatrix.c

```
#include<stdio.h>
void main()
{
        int i, j, m, n,flag=0;
        int a[5][5],b[5][5];
        printf("Enter the order of matrix : ");
        scanf("%d %d", &m, &n);
        printf("Enter %d elements : ", m*n);
        for ( i=0;i<m;i++)
        {
                for (j=0;j<n;j++ )
                {
                        scanf("%d", &a[i][j]);
                }
        }
        printf("The given matrix is\n");
        for ( i=0;i<m;i++)
        {
                for (j=0;j<n;j++ )
                {
                        printf("%d ", a[i][j]);
                }
                printf("\n");
        }
        printf("Transpose of the given matrix is\n");
        for ( i=0;i<n;i++)
        {
                for (j=0;j<m;j++ )
                {
                        b[i][j]=a[j][i];
                        printf("%d ", a[j][i]);
                }
                printf("\n");
        }
        for ( i=0;i<n;i++)
        {
                for (j=0;j<m;j++ )
                {
                        if(a[i][j]!=b[i][j])
                        {
                                flag=1;
                                break;
                        }
                }
        }
        if(flag==0)
        {
                printf("The given matrix is symmetric matrix\n");
        }
        else
        {
                printf("The given matrix is not symmetric matrix\n");
        }
}
```

## Execution Results - All test cases have succeeded!

| Test Case - 1 |
|---|
| **User Output** |
| Enter the order of matrix : |
| **2 2** |
| Enter 4 elements : |
| **1 2 3 4** |
| The given matrix is |
| 1 2 |
| 3 4 |
| Transpose of the given matrix is |
| 1 3 |
| 2 4 |
| The given matrix is not symmetric matrix |

| Test Case - 2 |
|---|
| **User Output** |
| Enter the order of matrix : |
| **2 2** |
| Enter 4 elements : |
| **4 5 5 4** |
| The given matrix is |
| 4 5 |
| 5 4 |
| Transpose of the given matrix is |
| 4 5 |
| 5 4 |
| The given matrix is symmetric matrix |

| Test Case - 3 |
|---|
| **User Output** |
| Enter the order of matrix : |
| **3 2** |
| Enter 6 elements : |
| **1 2 3 4 5 6** |
| The given matrix is |
| 1 2 |
| 3 4 |
| 5 6 |
| Transpose of the given matrix is |
| 1 3 5 |
| 2 4 6 |
| The given matrix is not symmetric matrix |

| Test Case - 4 |
| --- |
| **User Output** |
| Enter the order of matrix : |
| **3 3** |
| Enter 9 elements : |
| **111111111** |
| The given matrix is |
| 1 1 1 |
| 1 1 1 |
| 1 1 1 |
| Transpose of the given matrix is |
| 1 1 1 |
| 1 1 1 |
| 1 1 1 |
| The given matrix is symmetric matrix |

| S.No: 78 | Exp. Name: ***Write the code to determine whether the character is a vowel or consonant*** | Date: 2023-11-11 |
|---|---|---|



**Aim:**

**Write a program to determine whether the input character is a "vowel" or "consonant". Print "Invalid" if the given input is not an alphabet.**

**Constraint: The input can be either capital or small letter of alphabets.**

**Input and Output Format:**

- **Input consists of a single character.**
- **The output consists of a string "Vowel" or "Consonant" or "Invalid"**

**Instruction: To run your custom test cases strictly map your input and output layout with the visible test cases.**

**Source Code:**

vowelsCheck.c

```
//Write the code here
#include <stdio.h>

int isAlphabet(char ch) {
    return ((ch >= 'a' && ch <= 'z') || (ch >= 'A' && ch <= 'Z'));
}

int isVowel(char ch) {
    ch = tolower(ch); // Convert to lowercase for easier comparison
    return (ch == 'a' || ch == 'e' || ch == 'i' || ch == 'o' || ch == 'u');
}

int main() {
    char input;

    printf("Enter a character : ");
    scanf(" %c", &input);

    if (isAlphabet(input)) {
        if (isVowel(input))
            printf("Vowel\n");
        else
            printf("Consonant\n");
    } else {
        printf("Invalid\n");
    }

    return 0;
}
```

## Execution Results - All test cases have succeeded!

**Test Case - 1**

| User Output |
| --- |
| Enter a character : |
| **U** |
| Vowel |

| Test Case - 2 |
| --- |
| **User Output** |
| Enter a character : |
| **j** |
| Consonant |

| Test Case - 3 |
| --- |
| **User Output** |
| Enter a character : |
| **5** |
| Invalid |

| S.No: 79 | Exp. Name: ***Write a program to copy a string using string handling function (strcpy)?*** | Date: 2023-11-11 |
|---|---|---|

**Aim:**

**Write a program to copy a string using string handling function (strcpy)?**

**Source Code:**

stringcopy.c

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

int main() {
    char S1[100], S2[100];

    printf("Enter string for S2:");
    scanf("%s", S2);

    // Copy string from S2 to S1 using strcpy
    strcpy(S1, S2);

    printf("The String in S1: %s\n", S1);

    return 0;
}
```

## Execution Results - All test cases have succeeded!

| Test Case - 1 |
|---|
| **User Output** |
| Enter string for S2: |
| **kits** |
| The String in S1: kits |

| Test Case - 2 |
|---|
| **User Output** |
| Enter string for S2: |
| **code** |
| The String in S1: code |

| S.No: 80 | Exp. Name: ***Write the program to implement all string operations(copy,concatenate,compare,length) using string functions.*** | Date: 2023-11-11 |
|---|---|---|









**Aim:**

**Write the program to implement all string operations(copy,concatenate,compare,length) using string functions.**

**Source Code:**

string.c

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

int main() {
    char str1[100], str2[100], str3[200];

    printf("Enter string 1 :");
    scanf("%s", str1);

    printf("Enter string 2 :");
    scanf("%s", str2);

    // Length of strings
    printf("Length of string 1 is %d\n", (int)strlen(str1));
    printf("Length  of string 2 is %d\n", (int)strlen(str2));

    // Copy string 2 into another string
    strcpy(str3, str2);
    printf("After copying string 2 into another string:%s\n", str3);

    // Compare strings
    int compareResult = strcmp(str1, str2);
    if (compareResult == 0) {
        printf("After string comparing:Strings are equal\n");
    } else {
        printf("After string comparing:Strings are not equal\n");
    }

    // Concatenate strings
    strcat(str1, str2);
    printf("After string concatenate:%s\n", str1);

    return 0;
}
```

## Execution Results - All test cases have succeeded!

| Test Case - 1 |
|---|
| **User Output** |
| Enter string 1 : |

| |
|---|
| **welcome** |
| Enter string 2 : |
| **codetantra** |
| Length of string 1 is 7 |
| Length  of string 2 is 10 |
| After copying string 2 into another string:codetantra |
| After string comparing:Strings are not equal |
| After string concatenate:welcomecodetantra |

| Test Case - 2 |
|---|
| **User Output** |
| Enter string 1 : |
| **Apple** |
| Enter string 2 : |
| **Apple** |
| Length of string 1 is 5 |
| Length  of string 2 is 5 |
| After copying string 2 into another string:Apple |
| After string comparing:Strings are equal |
| After string concatenate:AppleApple |

| S.No: 82 | Exp. Name: ***Program to count the characters, words and lines in a text.*** | Date: 2023-11-11 |
|---|---|---|

**Aim:**

**Write a C program to count the characters, words and lines in a text.**

**Source Code:**

counts.c

```
#include <stdio.h>
#include <stdio.h>
#include <ctype.h>
#include <string.h>

int main() {
    char text[1000];
    int charCount = 0, wordCount = 0, lineCount = 0;
    int inWord = 0; // Flag to indicate if we are inside a word

    printf("Enter text (Type 'END' on a new line to finish input):\n");

    while (1) {
        fgets(text, sizeof(text), stdin);

        if (strcmp(text, "END\n") == 0) {
            break;  // End input when 'END' is entered on a new line
        }

        lineCount++; // Increment line count for each line of input

        for (int i = 0; text[i] != '\0'; i++) {
            char ch = text[i];
            charCount++;

            // Check for whitespace characters
            if (isspace(ch)) {
                inWord = 0; // We are not inside a word
            } else {
                // If not a whitespace character, we are inside a word
                if (!inWord) {
                    wordCount++;
                    inWord = 1; // Set the flag to indicate we are inside a word
                }
            }
        }
    }

    printf("Number of characters: %d\n", charCount);
    printf("Number of Words: %d\n", wordCount);
    printf("Number of lines: %d\n", lineCount);

    return 0;
}
```

## Execution Results - All test cases have succeeded!

| Test Case - 1 |
| --- |
| **User Output** |
| `Enter text (Type 'END' on a new line to finish input):` |
| **How is your experience** |
| **in codding this question** |
| **hope it is challenging you as you expected** |
| **END** |
| `Number of characters: 91` |
| `Number of Words: 16` |
| `Number of lines: 3` |

| Test Case - 2 |
| --- |
| **User Output** |
| `Enter text (Type 'END' on a new line to finish input):` |
| **Hello** |
| **Welcome to CodeTantra and** |
| **Ready to learn code with joy** |
| **end the input** |
| **by typing the word word in** |
| **capitals like as below** |
| **END** |
| `Number of characters: 125` |
| `Number of Words: 24` |
| `Number of lines: 6` |

| S.No: 83 | Exp. Name: ***Write a c program to implement string operations(length,copy,concatenate,compare,lowercase and uppercase) without using predefined string functions.*** | **Date: 2023-11-11** |
|---|---|---|

**Aim:**

**Write a c program to implement string operations(length,copy,concatenate,compare,lowercase and uppercase) without using predefined string functions.**

**Source Code:**

stringOperations.c

```c
#include <stdio.h>
int compare(str,str);
int main()
{
    char s1[100], s2[100], x[100], temp1[100];
    int i, j;
    printf("Enter string1: ");
    scanf("%s", s1);
    printf("Enter string2: ");
    scanf("%s", s2);

    for(i = 0; s1[i] != '\0'; ++i);
    printf("Length of string 1: %d\n", i);

    for(i = 0; s2[i] != '\0'; i++);
    printf("Length of string 2: %d\n", i);

    for(i = 0; s1[i] != '\0'; i++)
        temp1[i] = s1[i];
    temp1[i] = '\0';

    for(i=0; s2[i]!='\0'; i++)
        x[i] = s2[i];
    x[i] = '\0';

    printf("After copying string2: ");
    printf("%s\n",x);

    for(i = 0; s1[i] != '\0'; i++)
        x[i] = s1[i];
    for(j = 0; s2[j] != '\0'; j++)
        x[i+j] = s2[j];
    printf("After contatenating: ");
    printf("%s\n", x);
    printf("After comparing: ");
    int c = compare(s1, s2);
    if(c == 0)
        printf("strings are same\n");
    else
        printf("strings are not same\n");

    printf("After changing string 1 from lower to upper case: ");
    for (i = 0; s1[i] != '\0'; i++)
    {   if(s1[i] >= 'a' && s1[i] <= 'z')
            s1[i] = s1[i] - 32;
    }
    printf("%s\n", s1);

    printf("After changing string 1 from upper to lower case: ");
    for (i = 0; temp1[i] != '\0'; i++)
    {   if(temp1[i] >= 'A' && temp1[i] <= 'Z')
            temp1[i] = temp1[i] + 32;
    }
    printf("%s", temp1);
    return 0;
```

```
int compare(char a[],char b[])
{   int flag = 0, i = 0;
    while(a[i] != '\0' && b[i] != '\0')
    {  if(a[i] != b[i])
       {   flag = 1;
           break;
       }
       i++;
    }
    if(flag == 0)
        return 0;
    else
        return 1;
}
```



## Execution Results - All test cases have succeeded!

| Test Case - 1 |
|---|
| **User Output** |
| Enter string1: |
| **KITsW** |
| Enter string2: |
| **College** |
| Length of string 1: 5 |
| Length of string 2: 7 |
| After copying string2: College |
| After contatenating: KITsWCollege |
| After comparing: strings are not same |
| After changing string 1 from lower to upper case: KITSW |
| After changing string 1 from upper to lower case: kitsw |

| Test Case - 2 |
|---|
| **User Output** |
| Enter string1: |
| **WelCoMe** |
| Enter string2: |
| **CodeR** |
| Length of string 1: 7 |
| Length of string 2: 5 |
| After copying string2: CodeR |
| After contatenating: WelCoMeCodeR |
| After comparing: strings are not same |
| After changing string 1 from lower to upper case: WELCOME |
| After changing string 1 from upper to lower case: welcome |

| S.No: 84 | Exp. Name: ***Write a Simple C program using function for reading and printing the MATRIX*** | Date: 2023-11-11 |
|---|---|---|



**Aim:**

**Write a Simple C program using function for reading and printing the MATRIX**

**ALGORITHM:**

**Procedure for displaying elements**
**for i: =1 to n do**
**for j:=1 to m do**
**Read the elements into matrix by scanf() function.**
**i: =i+1**
**j: =j+1**

**Procedure for displaying elements**
**for i:=1 to n do**
**for j: =1 to n do**
**Display the elements of matrix by printf() funtion.**
**i: =i+1**
**j: =j+1**

**Source Code:**

1.c

```
#include<stdio.h>
int main()
{
    int array1[10][10],i,j,m,n,sum = 0;

    printf("Enter no. of rows :: ");
        scanf("%d", &m);
        printf("Enter no. of columns :: ");
        scanf("%d",&n);
        printf("Enter values to the matrix :: \n");
        for (i = 0; i < m; i++)
        {
            for (j = 0; j < n; j++)
            {
                 printf("Enter a[%d][%d] value :: ",i,j);
                 scanf("%d", &array1[i][j]);
        }
    }

    printf("The given matrix is \n");
        for (i = 0; i < m; ++i)
        {
        for (j = 0; j < n; ++j)
        {
            printf("\t%d", array1[i][j]);
        }
        printf("\n");
    }

    return 0;
}
```

## Execution Results - All test cases have succeeded!

| Test Case - 1 |
|---|
| **User Output** |
| Enter no. of rows :: |
| **2** |
| Enter no. of columns :: |
| **3** |
| Enter values to the matrix :: |
| Enter a[0][0] value :: |
| **34** |
| Enter a[0][1] value :: |
| **5** |
| Enter a[0][2] value :: |
| **56** |
| Enter a[1][0] value :: |
| **7** |
| Enter a[1][1] value :: |
| **5** |
| Enter a[1][2] value :: |
| **43** |
| The given matrix is |
| 34 5 56 |
| 7 5 43 |

| Test Case - 2 |
|---|
| **User Output** |
| Enter no. of rows :: |
| **4** |
| Enter no. of columns :: |
| **5** |
| Enter values to the matrix :: |
| Enter a[0][0] value :: |
| **12** |
| Enter a[0][1] value :: |
| **45** |
| Enter a[0][2] value :: |
| **1** |
| Enter a[0][3] value :: |
| **45** |
| Enter a[0][4] value :: |
| **2** |
| Enter a[1][0] value :: |
| **3** |
| Enter a[1][1] value :: |
| **67** |
| Enter a[1][2] value :: |

```
98
Enter a[1][3] value ::
32
Enter a[1][4] value ::
56
Enter a[2][0] value ::
34
Enter a[2][1] value ::
69
Enter a[2][2] value ::
90
Enter a[2][3] value ::
56
Enter a[2][4] value ::
34
Enter a[3][0] value ::
342
Enter a[3][1] value ::
67
Enter a[3][2] value ::
34
Enter a[3][3] value ::
23
Enter a[3][4] value ::
67
The given matrix is
        12      45      1       45      2
        3       67      98      32      56
        34      69      90      56      34
        342     67      34      23      67
```

| S.No: 85 | Exp. Name: ***Simple program using function for addition of two matrices*** | Date: 2023-11-11 |
|---|---|---|

**Aim:**

**Simple program using function for addition of two matrices**

**ALGORITHM:**

**Steps:**
**1.Start**
**2.Declare three array variable a,b,i,j.**
**3.Read the number of rows and columns for the first matrix i.e. m, n.**
**For i=1 to m do**
**For j=1 to n do**
**Read the element**
**j=j+ 1**
**j=j+ 1**
**4.Read the number of rows and columns for the second matrix i.e. p, q.**
**For i=1 to p do**
**For j=1 to q do**
**read the element**
**j=j+ 1**
**j=j+ 1**
**5. if(m !=p and n !=q) Then**
**Display " Addition not possible"**
**else**
**Call add function**
**6. Display the elements of matrix c**
**7. Stop**

**Source Code:**

1.c

```c
#include <stdio.h>

int main() {
      int m, n, c, d, first[10][10], second[10][10], sum[10][10];

   printf("Enter the number of rows and columns of matrix\n");
   scanf("%d%d", &m, &n);
   printf("Enter the elements of first matrix\n");

  for (c = 0; c < m; c++)
      for (d = 0; d < n; d++)
         scanf("%d", &first[c][d]);

   printf("Enter the elements of second matrix\n");

   for (c = 0; c < m; c++)
      for (d = 0 ; d < n; d++)
         scanf("%d", &second[c][d]);

        // Display the result
       printf("Sum of entered matrices:-\n");

   for (c = 0; c < m; c++) {
      for (d = 0 ; d < n; d++) {
         sum[c][d] = first[c][d] + second[c][d];
         printf("%d\t", sum[c][d]);
      }
      printf("\n");
   }

   return 0;
}
```

## Execution Results - All test cases have succeeded!

| Test Case - 1 |
|---|
| **User Output** |
| Enter the number of rows and columns of matrix |
| **2** |
| **3** |
| Enter the elements of first matrix |
| **3** |
| **4** |
| **5** |
| **3** |
| **4** |
| **6** |
| Enter the elements of second matrix |
| **6** |
| **6** |

| |
|---|
| 7 |
| 4 |
| 6 |
| 8 |
| Sum of entered matrices:- |
| 9 10 12 |
| 7 10 14 |

| Test Case - 2 |
|---|
| User Output |
| Enter the number of rows and columns of matrix |
| 3 |
| 3 |
| Enter the elements of first matrix |
| 12 |
| 34 |
| 56 |
| 34 |
| 12 |
| 45 |
| 45 |
| 6 |
| 12 |
| Enter the elements of second matrix |
| 1 |
| 5 |
| 7 |
| 2 |
| 34 |
| 1 |
| 4 |
| 5 |
| 6 |
| Sum of entered matrices:- |
| 13 39 63 |
| 36 46 46 |
| 49 11 18 |

| S.No: 87 | Exp. Name: ***Swapping of two numbers using call by value.*** | Date: 2023-11-11 |
|---|---|---|



**Aim:**

**Swapping of two numbers using call by value.**

**ALGORITHM:**

1. **Start**
2. **Declare two variable a,b**
3. **Read the values into a,b.**
4. **Display "values before swapping"**
5. **Print the values of variables a,b**
6. **Call the swap function by passing a,b**
7. **Display "values after swapping".**
8. **Print the values of a,b.**
9. **Stop**

**Source Code:**

1.c

```
#include <stdio.h>

// Function to swap two numbers using call by value
void swap(int x, int y) {
    int temp;
    temp = x;
    x = y;
    y = temp;

    printf("Values after swapping : x = %d y = %d\n", x, y);
}

int main() {
    int a, b;

    // Input values for a and b
    printf("Enter a value:");
    scanf("%d", &a);

    printf("Enter b value:");
    scanf("%d", &b);

    // Display values before swapping
    printf("Values before swapping : a = %d b = %d\n", a, b);

    // Call the swap function
    swap(a, b);



    return 0;
}
```

## Execution Results - All test cases have succeeded!

| Test Case - 1 |
| --- |
| **User Output** |
| Enter a value: |
| **5** |
| Enter b value: |
| **6** |
| Values before swapping : a = 5 b = 6 |
| Values after swapping : x = 6 y = 5 |

| Test Case - 2 |
| --- |
| **User Output** |
| Enter a value: |
| **95** |
| Enter b value: |
| **84** |
| Values before swapping : a = 95 b = 84 |
| Values after swapping : x = 84 y = 95 |

| S.No: 88 | Exp. Name: ***Swapping of two numbers using call by reference.*** | Date: 2023-11-11 |
|---|---|---|



**Aim:**

**Swapping of two numbers using call by reference.**

**ALGORITHM:**

10. **Start**
11. **Declare two variable a,b**
12. **Read the values into a,b.**
13. **Display "values before swapping"**
14. **Print the values of variables a,b**
15. **Call the swap function by passing &a, &b**
16. **Display "values after swapping".**
17. **Print the values of a,b.**
18. **Stop**

**Source Code:**

1.c

```
#include <stdio.h>

// Function to swap two numbers using call by reference
void swap(int *x, int *y) {
    int temp;
    temp = *x;
    *x = *y;
    *y = temp;

    printf("Values in the swap function : x = %d  y = %d\n", *x, *y);
}

int main() {
    int a, b;

    // Input values for a and b
    printf("Enter the value of x and y\n");
    scanf("%d %d", &a, &b);

    // Display values before swapping
    printf("Values before Swapping\n");
    printf("x = %d\ny = %d\n", a, b);

    // Call the swap function by passing references
    swap(&a, &b);

    // Display values after swapping
    printf("Values after Swapping\n");
    printf("x = %d\ny = %d\n", a, b);

    return 0;
}
```

## Execution Results - All test cases have succeeded!

| Test Case - 1 |
| --- |
| **User Output** |
| Enter the value of x and y |
| **4** |
| **6** |
| Values before Swapping |
| x = 4 |
| y = 6 |
| Values in the swap function : x = 6  y = 4 |
| Values after Swapping |
| x = 6 |
| y = 4 |

| Test Case - 2 |
| --- |
| **User Output** |
| Enter the value of x and y |
| **45** |
| **18** |
| Values before Swapping |
| x = 45 |
| y = 18 |
| Values in the swap function : x = 18  y = 45 |
| Values after Swapping |
| x = 18 |
| y = 45 |