

Rajiv-0920 / C-Programming

Code Issues Pull requests Actions Projects Wiki Security Insights Set

C-Programming / README.md



Rajiv-0920 initial changes

5 minutes ago



8578 lines (5763 loc) · 129 KB

Preview Code Blame

Raw



## Basic Simple C Programs

### 1. WAP(Write a program) to print "Hello World".

Expected Output

Hello World



Source Code

```
#include <stdio.h>

int main(){
    printf("Hello World");
    return 0;
}
```

### 2. WAP. to input a number and print them.

Test Data

Enter a number: 5



Expected Output

Number is 5



Source Code

```
#include <stdio.h>

int main(){
    int num;
```

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

    printf("Number is %d", num);

    return 0;
}
```

### 3. WAP. to accept two numbers and print their addition, subtraction, multiplication, division. ↗

Test Data

```
Enter 2 number: 5 10
```



Expected Output

```
Addition = 15
Subtraction = -5
Multiplication = 50
Division = 0
```



Source Code

```
#include <stdio.h>

int main(){

    int add, mul, sub, div, num1, num2;

    printf("\nEnter 2 number:");
    scanf("%d %d", &num1, &num2);

    add = num1 + num2;
    sub = num1 - num2;
    mul = num1 * num2;
    div = num1 / num2;

    printf("Addition = %d \nSubtraction = %d \nMultiplication = %d \nDivision = %d"
        , add, sub, mul, div);

    return 0;
}
```

### 4. WAP. to input a number and change the sign. ↗

Test Data

```
Enter a number: 5
```



Enter a number: -5

#### Expected Output

Changed number = -5  
Changed number = 5



#### Source Code

```
#include <stdio.h>

int main(){
    int num;

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

    num = num * -1;

    printf("Changed number = %d", num);
    return 0;
}
```



## 5. WAP. to input two number and display quotient and remainder. ↗

#### Test Data

input number : 98 4



#### Expected Output

quotient = 24  
remainder = 2



#### Source Code

```
#include <stdio.h>

int main(){
    int dividend, divisor, quotient, remainder;

    printf("Enter Dividend and Division: ");
    scanf("%d %d", &dividend, &divisor);

    quotient = dividend / divisor;
    remainder = dividend % divisor;

    printf("Quotient = %d Remainder = %d", quotient, remainder);

    return 0;
}
```



## 6. WAP. to display last digit of a number. ↴

Test Data

Enter a number: 153



Expected Output

Last digit = 3



Source Code

```
#include <stdio.h>

int main(){
    int num;

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

    num = num % 10;

    printf("Last digit = %d", num);

    return 0;
}
```



## 7. WAP. to accept a number from user and print it's square & cube in C language.



Test Data

Enter a number = 5



Expected Output

Square = 25 Cube = 125



Source Code

```
#include <stdio.h>

int main(){

    int num, cube, square;

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



```

        cube = num * num * num;
        square = num * num;

        printf("Square: %d Cube: %d \n", square, cube);

        return 0;
}

```

## 8. WAP. to calculate Area and Circumference of a Circle. ↗

### Formula

**Area of a Circle** =  $\pi r^2$

**Circumference of a circle** =  $2\pi r$

### Test Data

Enter Radius: 15



### Expected Output

Area of a circle = 78.525002  
 Circumference of a circle = 31.410000



### Source Code

```

#include <stdio.h>

int main(){

    float area, circum, radius, pi = 3.14153;

    printf("\nEnter Radius: ");
    scanf("%f", &radius);

    area = radius * radius * pi;
    circum = 2 * pi * radius;

    printf("Area of the circle: %f \n", area);
    printf("Circumference of the circle: %f \n", circum);

    return 0;
}

```

## 9. WAP. to input a number to compute the perimeter and area of a rectangle. ↗

### Formula

**Perimeter of the rectangle** =  $2(\text{height} + \text{width})$

**Area of Rectangle** =  $\text{height} * \text{width}$

#### Test Data

Enter height and width of the rectangle respectively: 12 5



#### Expected Output

Area of a rectangle = 60 square inches  
Perimeter of a rectangle = 34 inches



#### Source Code

```
#include <stdio.h>

int main(){
    int height, width, area, perimeter;

    printf("Enter height and width of the rectangle respectively: ");
    scanf("%d %d", &height, &width);

    area = height * width;
    perimeter = 2 * (height + width);

    printf("Area of a rectangle = %d square inches
\nPerimeter of a rectangle = %d inches", area, perimeter);

    return 0;
}
```



## 10. WAP. to Calculate Percentage of 5 Subjects. ↗

#### Test Data

Enter marks of 5 subjects:72 93 56 80 57



#### Expected Output

Your Overall Percentage: 71.599998



#### Source Code

```
#include <stdio.h>

int main(){

    float sanskrit, math, eng, hin, accounts, percentage, total;

    printf("Enter marks of 5 subjects:");
    scanf("%f %f %f %f %f", &sanskrit, &hin, &eng, &math, &accounts);

    total = sanskrit + hin + eng + math + accounts;
    percentage = total / 500 * 100;
}
```



```
    printf("\nTotal Marks = %f \nYour Overall Percentage: %f", total,  
percentage);  
  
    return 0;  
}
```

## 11. WAP. to Calculate Simple Interest. ↗

### Formula

$$\text{Simple Interest} = (p * r * t) / 100;$$

$p$  = Principal,  $r$  = Rate of interest,  $t$  = Time period

### Test Data

```
Enter Principal Amount: 4500  
Enter Rate of Interest: 9.5  
Enter Time: 6
```



### Expected Output

```
Simple interest: 2565.000000
```



### Source Code

```
#include <stdio.h>  
  
int main(){  
  
    float si, amount, interest, time;  
  
    printf("Enter Principal Amount: ");  
    scanf("%f", &amount);  
  
    printf("Enter Rate of Interest: ");  
    scanf("%f", &interest);  
  
    printf("Enter Time: ");  
    scanf("%f", &time);  
  
    si = (amount * interest * time) / 100;  
  
    printf("Simple interest: %f", si);  
  
    return 0;  
}
```



## 12. WAP. to print area of a triangle. ↗

### Formula

**Triangle** =  $0.5 * \text{Base} * \text{Height}$

Test Data

Enter BASE and HEIGHT: 15 30



Expected Output

Area of Triangle : 225.000000



Source Code

```
#include <stdio.h>

int main(){
    float area, base, height;

    printf("Enter BASE and HEIGHT: ");
    scanf("%f %f", &base, &height);

    area = 0.5 * base * height;

    printf("Area of Triangle : %f", area);

    return 0;
}
```



**13. WAP. to accept marks of 3 subjects of a student, Calculate total of 3 subjects and average in c language** ↗

Formula

**Average** = Sanskrit + Hindi + Math/3

Test Data

Enter Marks of 3 subjects: 75 50 80



Expected Output

Total marks: 205.000000  
Average marks: 68.333336



Source Code

```
#include <stdio.h>

int main(){
    float sub1, sub2, sub3, average, total;
```



```
printf("Enter marks of 3 subjects: ");
scanf("%f %f %f", &sub1, &sub2, &sub3);

total = (sub1 + sub2 + sub3);
average = total / 3;

printf("\nTotal marks: %f", total);
printf("\nAverage marks: %f", average);

return 0;
}
```

#### 14. WAP. to input paisa and convert it into rs. and paisa ↗

Test Data

Enter paisa:2150



Expected Output

Total 21 ₹ and 50 Paisa



Source Code

```
#include <stdio.h>

int main(){
    int paisa, rs;

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

    rs = paisa / 100;
    paisa = paisa % 100;

    printf("Total %d ₹ and %d Paisa", rs, paisa);

    return 0;
}
```

#### 15. WAP. to print the following outputs: <https://www.google.com> in C language ↗

Expected Output

Google Link: <https://www.google.com>



Source Code

```
#include <stdio.h>

int main(){
    printf("Google Link: https:\\\\www.google.com\\");
    return 0;
}
```



## 16. WAP. For Converting Temperature Celsius Into Fahrenheit and Fahrenheit to Celsius ↗

Formula

Fahrenheit =  $((9/5) * c) + 32$  // or you can use 1.8 in place of 9/5

celsius =  $(f - 32) \times 5/9$

Test Data

```
Enter Celsius or Fahrenheit: 55
```



Expected Output

```
Celsius to Fahrenheit: 131.000000
Fahrenheit to Celsius: 12.777778
```



Source Code

```
#include <stdio.h>

int main(){
    float celFah, fahrenheit, celsius;

    printf("\nEnter Celsius or Fahrenheit: ");
    scanf("%f", &celFah);

    fahrenheit = (9.0 / 5.0 * celFah) + 32.0;
    celsius = (celFah - 32.0) * (5.0 / 9.0);

    printf("\nCelsius to Fahrenheit: %f", fahrenheit);
    printf("\nFahrenheit to Celsius: %f", celsius);

    return 0;
}
```



## 17. WAP. to Calculate Gross Salary of an Employee whose dearness allowance is 40% of basic salary and house rent allowance is 20% of basic salary. ↗

Formula

**Gross Salary =  $b + da + o$**

**b = Basic Salary, da = Dearness Allowance o = Other Allowance**

Test Data

Enter Basic Salary: 20000



Expected Output

Gross Salary = 32000



Source Code

```
#include <stdio.h>

int main(){
    int gs, bs, da, hra;

    printf("Enter Basic salary: ");
    scanf("%d", &bs);

    da = bs * 40 / 100;
    hra = bs * 20 / 100;
    gs = bs + da + hra;

    printf("Gross Salary = %d \n", gs);

    return 0;
}
```



**18. WAP. to print profit and profit percentage. Selling price and cost price is given by user.** >

Formula

**Profit = selling - cost**

**Profit Percentage =  $\frac{profit}{cost} * 100$**

Test Data

Enter Selling price and Cost price respectively: 200 150



Expected Output

Total Profit = 50% and Profit percentage = 33%

Source Code

```
int main(){
    int profit, profitPercentage, sellingPrice, costPrice;

    printf("Enter Selling price and Cost price respectively: ");
    scanf("%d %d", &sellingPrice, &costPrice);

    profit = sellingPrice - costPrice;
    profitPercentage = (profit * 100) / costPrice;

    printf("Total Profit = %d% and Profit percentage = %d%",
           profit, profitPercentage);

    return 0;
}
```

## 19. WAP. to calculate the remainder of 2 numbers without using % operator. ↗

Test Data:

```
Enter 2 number: 10 5
```

Expected Output:

```
Remainder = 0
```

Source Code

```
#include <stdio.h>

int main(){
    int divisor, dividend, remainder, quotient;

    printf("Enter 2 dividend and divisor: ");
    scanf("%d %d", &dividend, &divisor);

    remainder = dividend - divisor * (dividend / divisor);

    printf("\nRemainder = %d", remainder);

    return 0;
}
```

## 20. WAP. that accepts two item's weight (floating points' values ) and number of purchase (floating points' values) and calculate the average value of the items. ↗

Test Data :

```
Weight - Item1: 15
No. of item1: 5
```

```
Weight - Item2: 25  
No. of item2: 4
```

Expected Output:

```
Average Value = 19.444444
```



Source Code

```
#include <stdio.h>

int main(){
    float weight1, weight2, itemNum1, itemNum2, average;

    printf("Weight - Item1: ");
    scanf("%f", &weight1);

    printf("No. of Item1: ");
    scanf("%f", &itemNum1);

    printf("Weight - Item2: ");
    scanf("%f", &weight2);

    printf("No. of Item1: ");
    scanf("%f", &itemNum2);

    average = (weight1 * itemNum1 + weight2 * itemNum2) / (itemNum1 + itemNum2);

    printf("Average value of the item = %f", average);

    return 0;
}
```



## 21. WAP. to show swap of two numbers. ↗

- i) using three variable
- ii) without using third variable.
- iii) swap within a single line.



Test Data :

```
Input two number a and b: 5 10
```



Expected Output:

```
a = 10 and b = 5
```



Source Code

```

#include <stdio.h>

int main(){
    int a, b, temp;

    printf("Enter two number a and b:");
    scanf("%d %d", &a, &b);

    // swap two number using third variable.
    // temp = a;
    // a = b;
    // b = temp;

    // Swap two number without using third variable.
    // a = a + b;
    // b = a - b;
    // a = a - b;

    // Swap two number within single line.
    b = a + b - (a = b);

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

    return 0;
}

```

## 22. WAP. to SWAP three numbers (any format). ↗

- i) using four variable
- ii) without using four variable.
- iii) swap within a single line.

Test Data:

Enter 3 number : 5 10 15

Expected Output:

changed number = 15 5 10

Source Code

```

#include <stdio.h>

int main(){
    int a, b, c, temp;

    printf("Enter 3 number a, b and c: ");
    scanf("%d %d %d", &a, &b, &c);

    // using four variable

```

```

// temp = a;
// a = b;
// b = c;
// c = temp;

// Without using four variable.
// a = a + b + c;
// b = a - b - c;
// c = a - b - c;
// a = a - b - c;

// Swap numbers within a single line.
a = (temp = a + b + c) - (b = temp - b - c) -
(c = temp - b - c);

printf("Changed number = %d %d %d", a, b, c);

return 0;
}

```

### 23. WAP. to merge three number. E.g. a= 1, b= 2, c = 8 is 128. ↗

| Test Data:

```
Enter 3 number: 1 2 3
```



| Expected Output:

```
merge number = 123
```



| Source Code

```

#include <stdio.h>

int main(){
    int a, b, c, merge;

    printf("Enter 3 number:\n");
    scanf("%d %d %d", &a, &b, &c);

    merge = a * 10;
    merge = merge + b;
    merge = merge * 10;
    merge = merge + c;

    printf("merge number = %d", merge);

    return 0;
}

```

## 24 .WAP. to Print the range of a number. E.g. number 78 is between 70 and 79, 102 is between 100 and 109. ↗

Test Data:

```
Enter a number : 78
```

```
Enter a number : 102
```



Expected Output:

```
78 number is between 70 and 79
```



```
102 number is between 100 and 109
```

Source Code

```
#include <stdio.h>

int main(){
    int num, x, y;

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

    x = num / 10 * 10;
    y = x + 9;

    printf("%d number is between %d and %d", num, x, y);

    return 0;
}
```



## 25. WAP. to input a 3 digit number and reverse it. ↗

Test Data:

```
number = 123
```



Expected Output:

```
reverse number = 321
```



Source Code

```
#include <stdio.h>

int main(){
    int num, rev, rem;

    printf("Enter a 3 digit number: ");
```



```
scanf("%d", &num);

rem = num % 10;
num = num / 10;
rev = rev * 10 + rem;
rem = num % 10;
num = num / 10;
rev = rev * 10 + rem;
rem = num % 10;
num = num / 10;
rev = rev * 10 + rem;

printf("Reverse number = %d\n", rev);

return 0;
}
```

## 26. WAP. to calculate sum of the digits of three digit number. ↗

Test Data

Enter a 3 digit number: 123



Expected Output

Sum = 6



Source Code

```
#include <stdio.h>

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

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

    sum = sum + num % 10;
    num = num / 10;
    sum = sum + num % 10;
    sum = sum + num / 10;

    printf("Sum = %d", sum);

    return 0;
}
```

## 27. WAP. to input a 5 digit number and calculate the sum of last and first digit number. ↗

Test Data:

number : 12345



Expected Output:

sum = 6



Source Code

```
#include <stdio.h>

int main(){
    int num, sum;

    printf("Enter a 5 digit number: ");
    scanf("%d", &num);

    sum = (num % 10) + (num / 10000);

    printf("Sum = %d", sum);

    return 0;
}
```



## 28. WAP. to convert specified days into years, weeks and days. ↗

*Note: Ignore leap year.*

Test Data

Enter number of days: 415



Expected Output

Years = 1 Weeks = 7 Days = 1



Source Code

```
#include <stdio.h>

int main(){
    int days, years, weeks;

    printf("Enter number of days: ");
    scanf("%d", &days);

    years = days / 365;
    weeks = (days % 365) / 7;
    days = (days % 365) % 7;

    printf("Years = %d Weeks = %d Days = %d", years, weeks, days);
}
```



```
    return 0;  
}
```

## 29. WAP. to convert a given integer (in seconds) to hours, minutes and seconds.

Test Data :

Input seconds: 25300 

Expected Output:

There are:  
H:M:S - 7:1:40 

Source Code

```
#include <stdio.h>  
  
int main(){  
    int seconds, minutes, hours;  
  
    printf("Enter seconds: ");  
    scanf("%d", &seconds);  
  
    hours = seconds / 3600;  
    seconds = seconds % 3600;  
    minutes = seconds / 60;  
    seconds = seconds % 60;  
  
    printf("\nThere are:\nH:M:S - %d:%d:%d", hours, minutes, seconds);  
  
    return 0;  
}
```

## 30. WAP. to convert a given integer (in millimeters) to kilometers, meters and centimeters.

1 centimeter = 10 millimeters.  
1 meter = 100 centimeters.  
1 meter = 1,000 millimeters.  
1 kilometer = 1,000 meters. 

Test Data :

Input millimeters: 2535220 

Expected Output:

2.53 kilometers  
2535.22 Meters  
253522.0 Centimeters



#### Source Code

```
#include <stdio.h>

int main(){
    float millimeters, kilometers, meters, centimeters;

    printf("Enter millimeters:");
    scanf("%f", &millimeters);

    centimeters = millimeters / 10;
    meters = centimeters / 100;
    kilometers = meters / 1000;

    printf("Kilometers = %f Meters = %f Centimeters = %f",
        kilometers, meters, centimeters);

    return 0;
}
```



### 31. WAP. to read an amount (integer value) and break the amount into smallest possible number of bank notes. ↴

#### Test Data :

Input the amount: 375



#### Expected Output:

There are:  
3 Note(s) of 100.00  
1 Note(s) of 50.00  
1 Note(s) of 20.00  
0 Note(s) of 10.00  
1 Note(s) of 5.00  
0 Note(s) of 2.00  
0 Note(s) of 1.00



#### Source Code

```
#include <stdio.h>

int main(){
    int amount, hundred, fifty, twenty, ten, five, two, one;

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

    hundred = amount / 100;
```



```

fifty = (amount % 100) / 50;
twenty = ((amount % 100) % 50) / 20;
amount = ((amount % 100) % 50) % 20;
ten = amount / 10;
amount = amount % 10;
five = amount / 5;
amount = amount % 5;
two = amount / 2;
amount = amount % 2;
one = amount;

printf("\n%d Note(s) of 100.00", hundred);
printf("\n%d Note(s) of 50.00", fifty);
printf("\n%d Note(s) of 20.00", twenty);
printf("\n%d Note(s) of 10.00", ten);
printf("\n%d Note(s) of 5.00", five);
printf("\n%d Note(s) of 2.00", two);
printf("\n%d Note(s) of 1.00", one);

return 0;
}

```

## 32. Write a C program to calculate the distance between the two points

Formula

$$\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

Test Data :

```

Input x1: 25
Input y1: 15
Input x2: 35
Input y2: 10

```



Expected Output:

```
Distance between the said points: 11.1803
```



Source Code

```

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

int main(){
    float x1, x2, y1, y2, distance;

    printf("Enter x1, y1, x2 and y2 respectively:");
    scanf("%f %f %f %f", &x1, &y1, &x2, &y2);

    distance = sqrt(pow((x1 - x2), 2) + pow((y1 - y2), 2));

    printf("The distance between the two points = Square root(%f)", distance);
}

```

```
    return 0;  
}
```

### 33. WAP. to Print the Ascii Value of the Character. ↗

Test Data

Enter The Character: c



Expected Output

Value = 99



Source Code

```
#include <stdio.h>  
  
int main(){  
    char a;  
  
    printf("Enter The Character: ");  
    scanf("%c", &a);  
  
    printf("Value = %d", a);  
  
    return 0;  
}
```

### 34. WAP. to accept 3 characters and print the sum of their ascii.

Test Data

Enter 3 character: a b c



Expected Output

Sum of the 3 character = 294



Source Code

```
#include <stdio.h>  
  
int main(){  
    char a, b, c;  
    int sum;  
  
    printf("Enter 3 character: ");  
    scanf("%c %c %c", &a, &b, &c);
```

```

    sum = (int)a + (int)b + (int)c;

    printf("Sum of the 3 character = %d", sum);

    return 0;
}

```

### 35. WAP. to Display The Size of Different Data Types

Data Type	Size (bytes)	Range	Format Specifier
int	2	-2,147,483,648 to 2,147,483,647	%d
long int	4	-2,147,483,648 to 2,147,483,647	%ld
float	4	1.2E-38 to 3.4E+38	%f
double	8	1.7E-308 to 1.7E+308	%lf
long double	12	3.4E-4932 to 1.1E+4932	%Lf
char	1	-128 to 127	%c

#### Source Code

```

#include <stdio.h> 

```

## If/Else Statement

### 1. WAP. to print greatest among the two numbers.

### Test Data

Enter 2 Numbers: 15 20



### Expected Output

Greater number = 20



### Source Code

```
#include <stdio.h>

int main(){
    int num1, num2, great;

    printf("Enter 2 numbers: ");
    scanf("%d %d", &num1, &num2);

    if(num1 > num2)
        great = num1;
    else
        great = num2;

    printf("\nGreater number = %d", great);
    return 0;
}
```



## 2. WAP. to check the given number is positive or negative. ↗

### Test Data

Enter a number: 15



### Expected Output

Number is 'Positive'.



### Source Code

```
#include <stdio.h>

int main(){
    int num;

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

    if(num < 0)
        printf("number is 'Negative'.");
    else
        printf("number is 'Positive'.");
```



```
    return 0;  
}
```

### 3. WAP. to check the given number is even or odd. ↗

Test Data

Enter a number: 12



Expected Output

number is 'Even'.



Source Code

```
#include <stdio.h>  
  
int main(){  
    int num;  
  
    printf("\nEnter a number:");  
    scanf("%d", &num);  
  
    if(num % 2 == 0)  
        printf("\nnumber is 'Even'.");  
    else  
        printf("\nnumber is 'Odd'.");  
    return 0;  
}
```

### 4. WAP. to check the person is eligible for vote or not. ↗

Test Data

Enter your age: 17



Expected Output

Your are not eligible for vote.



Source Code

```
#include <stdio.h>  
  
int main(){  
    int age;  
  
    printf("\nEnter your age:");
```

```
scanf("%d", &age);

if(age < 18)
    printf("\nYour are not eligible for vote.");
else
    printf("\nYour are eligible for vote.");
return 0;
}
```

## 5. WAP. to input a character and check whether it is vowel or consonant. ↗

Test Data

```
Enter a character: a
```



Expected Output

```
Character is 'Vowel'.
```



Source Code

```
#include <stdio.h>

int main(){
    char ch;

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

    if(ch == 'a' || ch == 'e' || ch == 'i' || ch == 'o' || ch == 'u'
    || ch == 'A' || ch == 'E' || ch == 'I' || ch == 'O' || ch == 'U')
        printf("Character is 'Vowel'.");
    else
        printf("Character is 'Consonant'.");
}

return 0;
}
```

## 6. WAP. to check the given year is leap year or not. ↗

Test Data

```
Enter a year: 2023
```



Expected Output

```
Not a leap year
```



Source Code

```
#include <stdio.h>

int main(){
    int year;

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

    if(year % 4 == 0 && year % 100 != 0 || year % 400 == 0)
        printf("\nYear is Leap Year.");
    else
        printf("\nYear is Not a Leap Year.");

    return 0;
}
```

## 7. WAP. to check the given number is buzz number or not. ↗

Buzz number

Buzz numbers are those numbers that are *divisible by 7* or *end with 7*.

Test Data

```
Enter a number: 47
```

Expected Output

```
47 is a Buzz Number.
```

Source Code

```
#include <stdio.h>

int main(){
    int num;

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

    if(num % 7 == 0 || num % 10 == 7)
        printf("%d is a Buzz number.", num);
    else
        printf("%d is not a Buzz number.", num);

    return 0;
}
```

## 8. WAP. to input cost and sales price and calculate percentage of profit or loss. ↗

Test Data

Enter Cost price and Sales price : 20 45



#### Expected Output

Total Profit = 25 and Percentage of Profit = 125%



#### Source Code

```
#include <stdio.h>

int main(){
    int costPrice, sellingPrice, profit, perProfit, loss, perLoss;

    printf("Enter Cost price and Sales price: ");
    scanf("%d %d", &costPrice, &sellingPrice);

    if(costPrice > sellingPrice){
        loss = costPrice - sellingPrice;
        perLoss = loss * 100 / costPrice;
        printf("\nTotal Loss = %d₹ and Percentage of Loss = %d%%",
        loss, perLoss);
    } else{
        profit = sellingPrice - costPrice;
        perProfit = profit * 100 / costPrice;
        printf("\nTotal Profit = %d₹ and Percentage of Profit = %d%%",
        profit, perProfit);
    }

    return 0;
}
```



## 9. WAP. to input two number and check they are proper division or not. ↗

#### Test Data

Enter 2 numbers: 45 90



Enter 2 numbers: 45 5

#### Expected Output

They are not a proper division.



They are a proper division.

#### Source Code

```
#include <stdio.h>

int main(){
    int num1, num2;

    printf("\nEnter 2 number:");
```



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

if(num1 % num2 == 0)
    printf("\nThey are a proper division.");
else
    printf("\nThey are not a proper division.");
return 0;
}
```

10. WAP. to input age and marks. If age  $\geq 18$  and marks  $\geq 80$  then the student is eligible for admission otherwise not. ↗

Test Data

Enter age and marks: 18 85



Expected Output

You are eligible for Admission.



Source Code

```
#include <stdio.h>

int main(){
    int age, marks;

    printf("\nEnter your age and marks:");
    scanf("%d %d", &age, &marks);

    if(age >= 18 && marks >= 80)
        printf("\nYou are eligible for admission.");
    else
        printf("\nYou are not eligible for admission.");
    return 0;
}
```

11. WAP. to check the given number is perfect square or not. ↗

Test Data

Enter a number: 25



Expected Output

25 is a Perfect Square.



Source Code

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

int main(){
    int num, sqr;

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

    sqr = sqrt(num);

    if(sqr * sqr == num)
        printf("%d is a Perfect Square.", num);
    else
        printf("%d is not a Perfect Square.", num);

    return 0;
}
```

## 12. WAP. to check the given number is 3 digit number or not. ↗

Test Data

```
Enter a number: 22
```

Expected Output

```
22 is not a 3 digit number
```

Source Code

```
#include <stdio.h>

int main(){
    int num;

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

    if(num > 99 && num < 1000)
        printf("\n%d is a 3 digit number.", num);
    else
        printf("\n%d is not a 3 digit number.", num);

    return 0;
}
```

## 13. WAP. to calculate greatest among the 3 numbers. ↗

Test Data

Enter 3 numbers: 15 20 25



#### Expected Output

Greater number is 25



#### Source Code

```
#include <stdio.h>

int main(){
    int num1, num2, num3, great;

    printf("\nEnter 3 numbers:");
    scanf("%d %d %d", &num1, &num2, &num3);

    if(num1 > num2 && num1 > num3)
        great = num1;
    else if(num1 < num2 && num2 > num3)
        great = num2;
    else
        great = num3;

    printf("Greater number = %d", great);
    return 0;
}
```



## 14. WAP. to calculate the grade of a student after the input of marks of that student. ↗

#### Grade

Percentage  $\geq$  90 grade is 'A'

Percentage  $\geq$  70 grade is 'B'

Percentage  $\geq$  50 grade is 'C'

Percentage  $\geq$  35 grade is 'D'

Percentage  $<$  35 grade is 'F'

#### Test Data

Enter marks: 80



#### Expected Output

Grade is 'B'.



#### Source Code

```
#include <stdio.h>

int main(){
    int marks;

    printf("\nEnter marks:");
    scanf("%d", &marks);

    if(marks >= 90)
        printf("\nGrade is 'A'.");
    else if(marks >= 70)
        printf("\nGrade is 'B'.");
    else if(marks >= 50)
        printf("\nGrade is 'C'.");
    else if(marks >= 35)
        printf("\nGrade is 'D'.");
    else
        printf("\nGrade is 'F'.");
    return 0;
}
```

## 15. WAP. to Calculate Telephone Bill:- ↴

calls	Rate/call
First 50 calls	free
next 100 calls	₹3/call
next 200 calls	₹5/call
next 350 calls	₹7/call

### Test Data

Enter numbers of call: 155

### Expected Output

Total Telephone Bill = 325

### Source Code

```
#include <stdio.h>

int main(){
    int calls, totalBill;

    printf("\nEnter numbers of call:");
    scanf("%d", &calls);

    if(calls <= 50)
        totalBill = 0;
    else if(calls > 50 && calls < 150)
```

```

        totalBill = (calls - 50) * 3;
    else if(calls > 150 && calls < 350)
        totalBill = 300 + (calls - 150) * 5;
    else
        totalBill = 1300 + (calls - 350) * 7;

    printf("\nTotal Telephone Bill = %d", totalBill);

    return 0;
}

```

**16. WAP. to input 3 sides of a triangle and check whether it is possible or not. If possible then check whether the triangle is an equilateral, isosceles or scalene triangle.** ↗

#### Triangle

- **Equilateral Triangle:** A triangle is considered to be an equilateral triangle *when all three sides have the same length.*
- **Isosceles triangle:** *When two sides of a triangle are equal or congruent, then it is called an isosceles triangle.*
- **Scalene triangle:** *When none of the sides of a triangle are equal, it is called a scalene triangle.*

#### Test Data

Enter 3 sides of a triangle: 2 2 1



#### Expected Output

Isosceles Triangle.



#### Source Code

```

#include <stdio.h>

int main(){
    int s1, s2, s3;

    printf("Enter 3 sides of a triangle:");
    scanf("%d %d %d", &s1, &s2, &s3);

    if(s1 + s2 > s3 || s1 + s3 > s2 || s2 + s3 > s1){
        if(s1 == s2 && s2 == s3)
            printf("\nEquilateral Triangle.");
        else if(s1 != s2 && s2 != s3)
            printf("\nScalene Triangle.");
        else
            printf("\nIsosceles Triangle.");
    } else
        printf("\nNot a Valid Triangle.");
    return 0;
}

```

## 17. WAP. to compute income tax paid by an employee: ↴

Annual Salary	Rate of Income Tax
up to ₹100000	NO TAX
₹100001 to 150000	10% of amount exceeding ₹100000
₹150001 to ₹250000	₹5000 + 20% of the amount exceeding ₹150000
Above ₹250000	₹25000 + 30% of the amount exceeding ₹250000

Test Data

Enter your salary: 275000



Expected Output

Total tax paid by him = 32500.000000



Source Code

```
#include <stdio.h>

int main(){
    float salary, taxableAmount;

    printf("\nEnter your salary:");
    scanf("%f", &salary);

    if(salary < 100000)
        taxableAmount = 0;
    else if(salary > 100000 && salary <= 150000)
        taxableAmount = (salary - 100000) * 10 / 100;
    else if(salary > 150000 && salary <= 250000)
        taxableAmount = 5000 + (salary - 150000) * 20 / 100;
    else
        taxableAmount = 25000 + (salary - 250000) * 30 / 100;

    printf("\nTotal Tax paid by him = %f", taxableAmount);
    return 0;
}
```

## 18. WAP. to accept three numbers from user and print them in ascending and descending order in c. ↴

Test Data

Enter 3 numbers: 3 4 1



## Expected Output

```
Ascending order = 4 3 1
Descending order = 1 3 4
```



## Source Code

```
#include <stdio.h>

int main(){
    int num1, num2, num3;

    printf("\nEnter 3 numbers: ");
    scanf("%d %d %d", &num1, &num2, &num3);

    if(num1 > num2 && num1 > num3){
        if(num2 > num3){
            printf("\nAscending order: %d %d %d",
                num1, num2, num3);
            printf("\nDescending order: %d %d %d",
                num3, num2, num1);
        } else{
            printf("\nAscending order: %d %d %d",
                num1, num3, num2);
            printf("\nDescending order: %d %d %d",
                num2, num3, num1);
        }
    } else if(num2 > num1 && num2 > num3){
        if(num1 > num3){
            printf("\nAscending order = %d %d %d",
                num2, num1, num3);
            printf("\nDescending order = %d %d %d",
                num3, num1, num2);
        } else {
            printf("\nAscending order = %d %d %d",
                num2, num3, num1);
            printf("\nDescending order = %d %d %d",
                num1, num3, num2);
        }
    } else{
        if(num1 > num2){
            printf("\nAscending order = %d %d %d",
                num3, num1, num2);
            printf("\nDescending order = %d %d %d",
                num2, num1, num3);
        } else{
            printf("\nAscending order = %d %d %d",
                num3, num2, num1);
            printf("\nDescending order = %d %d %d",
                num1, num2, num3);
        }
    }

    return 0;
}
```



## 19. WAP. to check the given date is correct or not. ↗

Test Data

Enter Date month and year e.g.(dd mm yyyy): 29 2 2023



Expected Output

29-2-2023 is not a valid date.



Source Code

```
#include <stdio.h>

int main(){
    int date, month, year, valid = 0;

    printf("\nEnter Date month and year (dd mm yyyy):");
    scanf("%d %d %d", &date, &month, &year);

    if(year % 4 == 0 && year % 100 != 0 || year % 400 == 0){
        if(month == 2 && (date <= 29 && date >= 1)){
            valid = 1;
        } else {
            valid = 0;
        }
    } else {
        if(month == 2 && (date <= 28 && date >= 1)){
            valid = 1;
        } else {
            valid = 0;
        }
    }

    if(valid == 1 ||
    (((month == 1 || month == 3 || month == 5 || month == 7 || month == 8 ||
    month == 10 || month == 12) && (date <= 31 && date >= 1)) ||
    ((month == 4 || month == 6 || month == 9 || month == 11) &&
    (date <= 30 && date >= 1))))
        printf("\n%d-%d-%d is a Valid Date.", date, month, year);
    else
        printf("\n%d-%d-%d is not a Valid Date.", date, month, year);

    return 0;
}
```

## 20. WAP. to input week number and print weekday. ↗

Test Data

Enter Week number: 4



### Expected Output

Thursday



### Source Code

```
#include <stdio.h>

int main(){
    int weekNO;

    printf("\nEnter Week number:");
    scanf("%d", &weekNO);

    if(weekNO == 1)
        printf("\nMonday");
    else if(weekNO == 2)
        printf("\nTuesday");
    else if(weekNO == 3)
        printf("\nWednesday");
    else if(weekNO == 4)
        printf("\nThursday");
    else if(weekNO == 5)
        printf("\nFriday");
    else if(weekNO == 6)
        printf("\nSaturday");
    else if(weekNO == 7)
        printf("\nSunday");
    else
        printf("\nEnter a valid week number.");

    return 0;
}
```



## 21. WAP. to input month number and print number of days in that month. ↗

### Test Data

Enter month number: 4



### Expected Output

30 Days



### Source Code

```
#include <stdio.h>

int main(){
    int monthNum, flag;

    printf("\nEnter month number:");
    scanf("%d", &monthNum);
```



```

if(monthNum == 1 || monthNum == 3 || monthNum == 5 || monthNum == 7 ||
   monthNum == 8 || monthNum == 10 || monthNum == 12)
    flag = 1;
else if (monthNum == 2)
    flag = 3;
else if(monthNum == 4 || monthNum == 6
       || monthNum == 9 || monthNum == 11)
    flag = 2;
else
    flag = 0;

if(flag == 1)
    printf("\n31 Days");
else if(flag == 2)
    printf("\n30 Days");
else if(flag == 3)
    printf("\n28 or 29 Days");
else
    printf("\nNot a Valid month.");

return 0;
}

```

## 22. WAP. to check whether a triangle is valid or not, when the three angles of the triangle are entered through the keyboard. ↗

### Condition

A triangle is valid if the sum of all the three angles is *equal to 180 degrees*.

### Test Data

Enter 3 angles of a triangle: 90 45 45



### Expected Output

Valid Triangle



### Source Code

```

#include <stdio.h>

int main(){
    int a1, a2, a3;

    printf("\nEnter 3 angles of a triangle:");
    scanf("%d %d %d", &a1, &a2, &a3);

    if(a1 + a2 + a3 == 180)
        printf("\nValid Triangle");
    else
        printf("\nnot a Valid Triangle");
}

```

```
    return 0;
}
```

## 23. WAP. to print the second largest out of three numbers. ↗

Test Data

```
Enter 3 numbers: 45 99 53
```



Expected Output

```
Second largest number = 53
```



Source Code

```
#include <stdio.h>

int main(){
    int num1, num2, num3, secLargest;

    printf("Enter 3 numbers:");
    scanf("%d %d %d", &num1, &num2, &num3);

    if((num1 < num3 || num1 < num2) &&
       (num1 > num2 || num1 > num3))
        secLargest = num1;
    else if((num2 < num3 || num2 < num1) &&
            (num2 > num1 || num2 > num3))
        secLargest = num2;
    else
        secLargest = num3;

    printf("\nSecond Largest number = %d", secLargest);

    return 0;
}
```



## 24. WAP. to compute the pension of an employee. ↗

*If the person is male.*

Age	Pension
Age $\geq$ 90	pension is 4000
Age $\geq$ 60	pension is 6000
Age $<$ 60	pension is 0

*If the person is female.*

Age	Pension
Age >= 90	pension is 3000
Age >= 60	pension is 5000
Age < 60	pension is 0

#### Test Data

Enter Your Age and Gender: 99 f



#### Expected Output

Pension = 3000



#### Source Code

```
#include <stdio.h>

int main(){
    int age, pension;
    char gender;

    printf("\nEnter Your Age and Gender: ");
    scanf("%d %c", &age, &gender);

    if(age >= 90){
        if(gender == 'm')
            pension = 4000;
        else if(gender == 'f')
            pension = 3000;
        else
            printf("\nEnter a Valid Gender.");
    } else if(age >= 60){
        if(gender == 'm')
            pension = 6000;
        else if(gender == 'f')
            pension = 5000;
        else
            printf("\nEnter a Valid Gender.");
    } else {
        pension = 0;
    }

    printf("\nPension = %d", pension);
    return 0;
}
```

**25. WAP. to check whether a 3 digit number is a magic number or not.**  
**(Palindrome)** A number is a magic number if its reverse is same as the original number. ↗

#### Test Data

Enter number 121



Expected Output

121 is a magic number



Source Code

```
#include <stdio.h>

int main(){
    int num, rev, temp;

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

    temp = num;

    rev = num % 10;
    temp = temp / 10;
    rev = (rev * 10) + temp % 10;
    rev = (rev * 10) + temp / 10;

    printf("%d %d", rev, num);
    if(rev == num)
        printf("\n%d is a magic number.", num);
    else
        printf("\n%d is not a magic number.", num);

    return 0;
}
```

## 26. WAP. to Find the absolute value of a number entered through the keyboard.

Test Data

Enter a number: 11  
Enter a number: -23



Expected Output

Absolute number = 11  
Absolute number = 23



Source Code

```
#include <stdio.h>

int main(){
    int num;
```



```

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

if(num < 0)
    num = num * -1;

printf("Absolute number = %d", num);
return 0;
}

```

## 27. WAP. to accept users marital status, gender and age to check if he/she is eligible for marriage or not. ↗

Test Data

```

Enter MaritalStatus: m (married) / u (unmarried): u
Enter your gender: m (male) / f (female): m
Enter your age: 24

```



Expected Output

```
You can marry!
```



Source Code

```

#include <stdio.h>

int main(){

    int age;
    char maritalStatus, gender;

    printf("\nEnter MaritalStatus: m (married) / u (unmarried): ");
    scanf(" %c", &maritalStatus);

    printf("\nEnter your gender: m (male) / f (female): ");
    scanf(" %c", &gender);

    printf("\nEnter your age: ");
    scanf(" %d", &age);

    if(maritalStatus == 'm')
        printf("\nYou can not marry!");
    else if(maritalStatus == 'u'){
        if(gender == 'm'){
            if(age >= 21)
                printf("\nYou can marry!");
            else
                printf("\nYou can not marry!");
        } else if(gender == 'f'){
            if(age >= 18)
                printf("\nYou can marry!");
            else
                printf("\nYou can not marry!");
        } else
            printf("\nYou can not marry!");
    }
}

```

```
        printf("\nEnter valid gender: ");
    } else
        printf("\nEnter valid Marital Status: ");
    return 0;
}
```

## 28. WAP. to Count the total numbers of notes in given amount. ↗

### Test Data

Enter the amount: 375



### Expected Output

```
2000 = 0
500 = 2
200 = 1
100 = 0
50 = 0
20 = 1
10 = 0
5 = 0
2 = 0
1 = 1
```



### Source Code

```
#include <stdio.h>

int main(){
    int amount, twoThousand = 0, fiveHundred = 0, twoHundred = 0,
    oneHundred = 0, fifty = 0, twenty = 0, ten = 0,
    five = 0, two = 0, one = 0;

    printf("\nEnter amount:");
    scanf("%d", &amount);

    if(amount > 2000){
        twoThousand = amount / 2000;
        amount = amount % 2000;
    }

    if(amount > 500){
        fiveHundred = amount / 500;
        amount = amount % 500;
    }

    if(amount > 200){
        twoHundred = amount / 200;
        amount = amount % 200;
    }

    if(amount > 100){
        oneHundred = amount / 100;
        amount = amount % 100;
    }

    if(amount > 50){
        fifty = amount / 50;
        amount = amount % 50;
    }

    if(amount > 20){
        twenty = amount / 20;
        amount = amount % 20;
    }

    if(amount > 10){
        ten = amount / 10;
        amount = amount % 10;
    }

    if(amount > 5){
        five = amount / 5;
        amount = amount % 5;
    }

    if(amount > 2){
        two = amount / 2;
        amount = amount % 2;
    }

    if(amount > 1){
        one = amount / 1;
        amount = amount % 1;
    }
}
```



```

}

if(amount > 50){
    fifty = amount / 50;
    amount = amount % 50;
}

if(amount > 20){
    twenty = amount / 20;
    amount = amount % 20;
}

if(amount > 10){
    ten = amount / 10;
    amount = amount % 10;
}

if(amount > 5){
    five = amount / 5;
    amount = amount % 5;
}

if(amount > 2){
    two = amount / 2;
    amount = amount % 2;
}

if(amount >= 1){
    one = amount;
}

printf("\n2000 = %d", twoThousand);
printf("\n500 = %d", fiveHundred);
printf("\n200 = %d", twoHundred);
printf("\n100 = %d", oneHundred);
printf("\n50 = %d", fifty);
printf("\n20 = %d", twenty);
printf("\n10 = %d", ten);
printf("\n5 = %d", five);
printf("\n2 = %d", two);
printf("\n1 = %d", one);
return 0;
}

```

## 29. WAP. to determine whether the given character is a capital letter, a small case letter, a digit or a special symbol. ↗

Hint

**ASCII** value of the digit is between 48 to 58 and lowercase characters in the range of 97 to 122, and uppercase is between 65 and 90, and special symbol is between (32 to 47, 58 to 64, 91 to 96, 123 to 127).

Test Data

Enter Any Character: c



### Expected Output

Character is 'Lowercase'.



### Source Code

```
#include <stdio.h>

int main(){
    char ch;

    printf("Enter Any Character: ");
    scanf("%c", &ch);

    if(ch >= 48 && ch < 58){
        printf("Character is 'Digit'.");
    } else if(ch >= 32 && ch <= 47 || ch >= 58 && ch <= 64
    || ch >= 91 && ch <= 96 || ch >= 123 && ch <= 127){
        printf("Character is 'Special Character'.");
    } else if(ch >= 65 && ch <= 90){
        printf("Character is 'UpperCase'.");
    } else if(ch >= 97 && ch <= 121){
        printf("Character is 'Lowercase'.");
    }

    return 0;
}
```



30. WAP. to input the length and breadth of a rectangle, find whether the area of the rectangle is greater than its perimeter. ↗

### Test Data

Enter the length and breadth of a rectangle: 5 4



### Expected Output

Area of rectangle is greater than its perimeter.



### Source Code

```
#include <stdio.h>

int main(){
    int length, breadth, area, perimeter;

    printf("\nEnter the length and breadth of a rectangle:");
    scanf("%d %d", &length, &breadth);

    area = length * breadth;
    perimeter = 2 * (length + breadth);
```



```

    if(area > perimeter)
        printf("\nArea is greater than it's perimeter.");
    else
        printf("\nArea is not greater than it's perimeter.");

    return 0;
}

```

**31. WAP. to input three points  $(x_1, y_1)$ ,  $(x_2, y_2)$  and  $(x_3, y_3)$ , check if all the three points fall on one straight line.** ↴

Test Data

```

Enter points (x1, y1):1 2
Enter points (x2, y2):3 4
Enter points (x3, y3):5 6

```



Expected Output

```
All the three points fall on the straight line:
```



Source Code

```

#include <stdio.h>

int main(){
    int x1, y1, x2, y2, x3, y3, m, n;

    printf("Enter points (x1, y1):");
    scanf("%d %d", &x1, &y1);

    printf("Enter points (x2, y2):");
    scanf("%d %d", &x2, &y2);

    printf("Enter points (x3, y3):");
    scanf("%d %d", &x3, &y3);

    m = (y2 - y1) / (x2 - x1);
    n = (y3 - y2) / (x3 - x2);

    if(m == n)
        printf("All the three points fall on the straight line:");
    else
        printf("All 3 points do not lie on the same line\n");

    return 0;
}

```



**32. WAP. to input a point  $(x, y)$ , find out if it lies on the x-axis, y-axis or at the origin, viz.  $(0, 0)$ .** ↴

### Condition

In point  $(x, y)$ , if  $x = 0$  and  $y = 0$ , then the point lies on the origin. If value of  $x$  is zero and  $y$  is greater than zero, then the point lies on  $y$ -axis. If  $y$  is zero and  $x$  is greater than zero, then the point lies on  $x$ -axis.

### Test Data

Enter point  $(x, y)$ : 35 20



### Expected Output

Point (35, 20) neither lie on x-axis nor on y-axis



### Source Code

```
#include <stdio.h>

int main(){
    int x1, y1;

    printf("Enter (x, y): ");
    scanf("%d %d", &x1, &y1);

    if(x1 == 0 && y1 == 0)
        printf("\nThe point (%d, %d) lies on the origin.", x1, y1);
    else if(x1 == 0 && y1 > 0)
        printf("\nThe point (%d, %d) lies on the y-axis.", x1, y1);
    else if(x1 > 0 && y1 == 0)
        printf("\nThe point (%d, %d) lies on the x-axis.", x1, y1);
    else
        printf("Point (%d, %d) neither lie on x-axis nor on y-axis", x1, y1);
}
```

## 33. WAP. to check whether a given number is even or odd without using modulo (%) operator ↴

### Test Data:

Enter a number: 12



### Expected Output:

Even number



### Source Code

```
#include <stdio.h>

int main(){
    int num, rem;
```

```

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

rem = num - num / 2 * 2;

if(rem == 0)
    printf("\nEven number");
else
    printf("\nOdd number");

return 0;
}

```

### 34. WAP. An electricity board charges the following rates for the use of electricity:



Unit	Price
first 200 units	Rs. 0.80 per unit
next 100 units	Rs. 0.90 per unit
Beyond 300 units	Rs. 1 per unit

All users are charged a minimum of Rs. 100 as meter charge.

If the total amount is more than Rs. 400, then an additional surcharge of 15% of total amount is charged.

#### Test Data

Enter number of units: 417



#### Expected Output

Total Charges = 437.05



#### Source Code

```

#include <stdio.h>

int main(){
    float units, price;

    printf("\nEnter number of units: ");
    scanf("%f", &units);

    if(units <= 200 && units > 0)
        price = units * 0.80;
    else if(units <= 300 && units > 200)
        price = 160 + (units - 200) * 0.90;
    else if(units > 300)
        price = 250 + (units - 300) * 1;
    else
        price = 0;
}

```

```
    price = price + 100;

    if(price > 400)
        price = price + (price * 15 / 100);

    printf("\nTotal Charges = %.2f", price);

    return 0;
}
```

## Conditional or Ternary Operator ↗

### 1. WAP. to input a number and print even or odd. ↗

Test Data

```
Enter a number: 5
```



Expected Output

```
5 is a Odd number.
```



Source Code

```
#include <stdio.h>

int main(){
    int num;

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

    num % 2 == 0 ? printf("Even Number.") : printf("Odd number");

    return 0;
}
```

### 2. WAP. to find the greatest of the two numbers. ↗

Test Data

```
Enter two number: 5 10
```



Expected Output

Greater number = 10



#### Source Code

```
#include <stdio.h>

int main(){
    int num1, num2, great;

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

    great = num1 > num2 ? num1 : num2;

    printf("Greater number = %d", great);

    return 0;
}
```



### 3. WAP. to find the greatest of the three numbers. ↗

#### Test Data

Enter Three number: 22 43 10



#### Expected Output

Greater number = 43



#### Source Code

```
#include <stdio.h>

int main(){
    int num1, num2, num3, great;

    printf("\nEnter three number: ");
    scanf("%d %d %d", &num1, &num2, &num3);

    great = num1 > num2 ? num1 > num3 ?
        num1 : num3 : num2 > num3 ? num2 : num3;

    printf("\nGreater number = %d", great);

    return 0;
}
```



### 4. WAP. using conditional operators to determine whether a year entered through the keyboard is a leap year or not ↗

#### Test Data

Enter a year: 2021



#### Expected Output

Not a leap year.



#### Source Code

```
#include <stdio.h>

int main(){
    int year;

    printf("\nEnter a year:");
    scanf("%d", &year);

    year % 4 == 0 ? printf("\nLeap year.") :
    year % 100 != 0 && year % 400 == 0 ?
    printf("\nLeap year.") : printf("\nNot a Leap year.");

    return 0;
}
```



## 5. WAP. The cost of one type of mobile service is Rs. 250 plus Rs. 1.25 for each call made over and above 100 calls. print the bill. (ternary operator) ↗

#### Test Data

Enter number of Calls: 200



#### Expected Output

Total Bill = 251.25



#### Source Code

```
#include <stdio.h>

int main(){
    float calls, bill;

    printf("Enter number of Calls: ");
    scanf("%f", &calls);

    bill = calls > 100 ?(250 + (calls - 100) * 1.25 ): 250;

    printf("Total Bill = %.2f", bill);
```



```
    return 0;  
}
```

## 6. WAP. to find Whether the character entered through the keyboard is a lower case alphabet or uppercase. ↴

### Test Data

Enter a character: a



### Expected Output

Character is Lowercase.



### Source Code

```
#include <stdio.h>  
  
int main(){  
    char ch;  
  
    printf("Enter a character: ");  
    scanf("%c", &ch);  
  
    ch >= 65 && ch <= 90 ? printf("Character is Uppercase.") :  
    ch >= 97 && ch <= 121 ? printf("Character is Lowercase.") :  
    printf("Not a character.");  
  
    return 0;  
}
```

## Loop Control Instruction ↴

### 1. WAP. to print first 10 natural numbers. ↴

#### Natural numbers

First 10 Natural Numbers are: **1, 2, 3, 4, 5, 6, 7, 8, 9 and 10.**

#### Expected Output

1 2 3 4 5 6 7 8 9 10



#### Source Code

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



```
int main(){

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

    return 0;
}
```

## 2. WAP. to print first 10 natural numbers in reverse order. ↗

Natural numbers

First 10 Natural Numbers are: **1, 2, 3, 4, 5, 6, 7, 8, 9 and 10.**

Expected Output

```
10 9 8 7 6 5 4 3 2 1
```



Source Code

```
#include <stdio.h>

int main(){

    for(int i = 1; i >= 1; i--){
        printf("%d ", i);
    }

    return 0;
}
```



## 3. WAP. to print first 10 odd natural numbers ↗

Odd numbers

Odd numbers are **not multiples of 2.**

Expected Output

```
1 3 5 7 9 11 13 15 17 19
```



Source Code

```
#include <stdio.h>

int main(){

    int count = 0;

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

        if(i % 2 != 0){
            count++;

            if(count == 10){
                break;
            }
        }
    }

    return 0;
}
```



```
    if(i % 2 != 0){
        printf("%d ", i);
        count++;
    }
    if(count == 10)
        break;
}

return 0;
}
```

#### 4. WAP. to print first 10 even natural numbers ↗

Even numbers

Even numbers are *multiples of 2*.

Expected Output

2 4 6 8 10 12 14 16 18 20



Source Code

```
#include <stdio.h>

int main(){

    int count = 0;

    for(int i = 1; i<= 100; i++){
        if(i % 2 == 0){
            printf("%d ", i);
            count++;
        }
        if(count == 10)
            break;
    }

    return 0;
}
```



#### 5. WAP. to print first N natural numbers. Value of N is given by user. ↗

Test Data

Enter n number: 15



Expected Output

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15



## Source Code

```
#include <stdio.h>

int main(){
    int n;

    printf("\nEnter n number: ");
    scanf("%d", &n);

    for(int i = 1; i <= n; i++){
        printf("%d ", i);
    }
    return 0;
}
```



6. WAP. to print first N odd natural numbers in reverse order. Value of N is given by user. ↗

## Test Data

```
Enter n number: 25
```



## Expected Output

```
First 25 Odd numbers in reverse order = 25 23 21 19 17 15 13 11 9 7 5 3 1
```



## Source Code

```
#include <stdio.h>

int main(){
    int n;

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

    printf("\nFirst %d Odd numbers in reverse order = ", n);
    for(int i = n; i >= 1; i--){
        if(i % 2 != 0)
            printf("%d ", i);
    }

    return 0;
}
```



7. WAP. to calculate sum of first N natural numbers. Value of N is given by user. ↗

## Test Data

Enter n number: 5



#### Expected Output

Sum of 5 Natural number = 15



#### Source Code

```
#include <stdio.h>

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

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

    for(int i = 1; i <= n; i++){
        sum += i;
    }

    printf("\nSum of %d Natural number = %d", n, sum);
    return 0;
}
```



## 8. WAP to calculate factorial of a numbers. ↗

#### Factorial number

the factorial of 6 is  $1 * 2 * 3 * 4 * 5 * 6 = 720$

#### Test Data

Enter n number: 6



#### Expected Output

Factorial of 6 = 720



#### Source Code

```
#include <stdio.h>

int main(){
    int n, fact = 1;

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

    for(int i = n; i >= 1; i--){
        fact *= i;
    }
}
```



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

    return 0;
}
```

## 9. WAP. to calculate $x^y$ . Values of x and y are given by user. ↗

### Test Data

Enter x and y: 3 5



### Expected Output

Value of x and y = 243



### Source Code

```
#include <stdio.h>

int main(){
    int x, y, value = 1;

    printf("Enter x and y: ");
    scanf("%d %d", &x, &y);

    for(int i = 1; i <= y; i++){
        value *= x;
    }

    printf("Value of x and y = %d", value);
    return 0;
}
```

## 10. WAP. to count number of digits in a given number. ↗

### Test Data

Enter a number: 123



### Expected Output

3 number of digits in a given number.



### Source Code

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



```
int main(){
    int count = 0, num;

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

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

    printf("%d number of digits in a given number.", count);

    return 0;
}
```

## 11. WAP. to calculate sum of the digits of a given number. ↗

Test Data

Enter a number: 352



Expected Output

Sum = 10



Source Code

```
#include <stdio.h>

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

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

    while(num > 0){
        sum += num % 10;
        num /= 10;
    }

    printf("Sum = %d", sum);
    return 0;
}
```

## 12. WAP. to reverse a number. ↗

Test Data

Enter a number: 352



### Expected Output

```
Reverse number = 253
```



### Source Code

```
#include <stdio.h>

int main(){
    int num, rev = 0, rem;

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

    while(num > 0){
        rem = num % 10;
        rev = rev * 10 + rem;
        num /= 10;
    }

    printf("Reverse number = %d", rev);

    return 0;
}
```



## 13. WAP. to check whether the two given numbers are reverse of each other or not. ↴

### Test Data

```
Enter two number: 123 321
```



### Expected Output

```
YES
```



### Source Code

```
#include <stdio.h>

int main(){
    int num1, num2, rev = 0;

    printf("Enter a number: ");
    scanf("%d %d", &num1, &num2);

    while(num1 > 0){
        rev = rev * 10 + num1 % 10;
        num1 /= 10;
    }

    (rev == num2) ? printf("YES") : printf("NO");
}
```



```
    return 0;  
}
```

## 14. WAP. to check whether a given number is Prime or not. ↗

Test Data

Enter a number: 5



Expected Output

Prime number



Source Code

```
#include <stdio.h>  
  
int main(){  
    int num, flag = 1;  
  
    printf("Enter a number: ");  
    scanf("%d", &num);  
  
    for(int i = 2; i <= num / 2; i++){  
        if(num % i == 0){  
            flag = 0;  
            break;  
        }  
    }  
  
    if(num == 1)  
        flag = 0;  
  
    if(flag == 1)  
        printf("Prime number");  
    else  
        printf("Not a Prime number");  
    return 0;  
}
```



## 15. WAP. to print all prime numbers between two given numbers. ↗

Test Data

Enter the range between two number: 25 50



Expected Output

29 31 37 41 43 47



### Source Code

```
#include <stdio.h>

int main(){
    int n1, n2, flag;

    printf("Enter the range between two numbers: ");
    scanf("%d %d", &n1, &n2);

    printf("\nPrime numbers in the given range are = ");
    for(int i = n1; i <= n2; i++){
        flag = 1;
        for(int j = 2; j <= i / 2; j++){
            if(i % j == 0){
                flag = 0;
                break;
            }
        }
        if(i == 1)
            flag = 0;
        if(flag == 1)
            printf("%d ", i);
    }
    return 0;
}
```



## 16. WAP. to print all prime numbers in first 1000 natural numbers. ↗

### Source Code

```
#include <stdio.h>

int main(){
    int flag;

    printf("\nPrime numbers = ");
    for(int i = 1; i <= 1000; i++){
        flag = 1;
        for(int j = 2; j <= i / 2; j++){
            if(i % j == 0){
                flag = 0;
                break;
            }
        }
        if(i == 1)
            flag = 0;
        if(flag == 1)
            printf("%d ", i);
    }

    return 0;
}
```



## 17. WAP. to find GCD of two numbers. ↗

Greatest Common Divisor - GCD

Find the **GCD** of **15 and 20**.

```
Factor of 15 = 1, 3, 5, 15  
Factor of 20 = 1, 2, 4, 5, 10, 20
```

```
GCD of 15 and 20 = 5
```



Test Data

```
Enter two numbers: 36 60
```



Expected Output

```
GCD = 12
```



Source Code

```
#include <stdio.h>

int main(){
    int num1, num2, smaller, gcd;

    printf("\nEnter two numbers: ");
    scanf("%d %d", &num1, &num2);

    smaller = num1;

    if(num1 > num2)
        smaller = num2;

    while(smaller != 0){
        if(num1 % smaller == 0 && num2 % smaller == 0){
            gcd = smaller;
            break;
        }
        smaller--;
    }

    printf("\nGCD = %d", gcd);

    return 0;
}
```



## 18. WAP. to find LCM of two numbers. ↗

Test Data

```
Enter two number: 12 52
```



### Expected Output

```
LCM = 156
```



### Source Code

```
#include <stdio.h>

int main(){
    int num1, num2, greater, lcm;

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

    greater = num1;

    if(num1 < num2)
        greater = num2;

    while(1){
        if(greater % num1 == 0 && greater % num2 == 0){
            lcm = greater;
            break;
        }
        greater++;
    }

    printf("LCM = %d", lcm);

    return 0;
}
```



## 19. WAP. to print the following 1, 4, 7, 10 ..... 40. ↴

### Expected Output

```
1 4 7 10 13 16 19 22 25 28 31 34 37 40
```



### Source Code

```
#include <stdio.h>

int main(){
    int i = 1;

    while(i <= 40){
        printf("%d ", i);
        i += 3;
    }

    return 0;
}
```



## 20. WAP. to print the following 1, -4, 7, -10 ..... -40. ↴

Expected Output

```
1 -4 7 -10 13 -16 19 -22 25 -28 31 -34 37 -40
```



Source Code

```
#include <stdio.h>

int main(){
    int i = 1, s = 1;

    while(i <= 40){
        printf("%d ", i * s);
        i += 3;
        s *= -1;
    }

    return 0;
}
```



## 21. WAP. to print table of a given number. ↴

Test Data

```
Enter a number: 19
```



Expected Output

```
19 x 1 = 19
19 x 2 = 38
19 x 3 = 57
19 x 4 = 76
19 x 5 = 95
19 x 6 = 114
19 x 7 = 133
19 x 8 = 152
19 x 9 = 171
19 x 10 = 190
```



Source Code

```
#include <stdio.h>

int main(){
    int n, i = 1;

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



```
while(i <= 10){
    printf("\n%d x %d = %d", n, i, n * i);
    i++;
}

return 0;
}
```

## 22. WAP. to find a 3 digit number is Armstrong or not. (without using pow function) ↴

Armstrong number

0, 1, 153, 370, 371 and 407 are the Armstrong numbers.

$$153 = 1 * 1 * 1 + 5 * 5 * 5 + 3 * 3 * 3$$

$$153 = 1 + 125 + 27$$

$$153 = 153$$

Test Data

Enter a number: 153



Expected Output

153 is a Armstrong number



Source Code

```
#include <stdio.h>

int main(){
    int num, rem, arm = 0, temp;

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

    temp = num;

    while(temp != 0){
        rem = temp % 10;
        arm += rem * rem * rem;
        temp /= 10;
    }

    if(num == arm)
        printf("%d is a Armstrong number.", num);
    else
        printf("%d is not a Armstrong number.", num);
}
```

```
    return 0;
}
```

## 23. WAP. to Find Number Is Armstrong Or Not ↴

Armstrong number

0, 1, 153, 370, 371 and 407 are the Armstrong numbers.

$$153 = 1 * 1 * 1 + 5 * 5 * 5 + 3 * 3 * 3$$

$$153 = 1 + 125 + 27$$

$$153 = 153$$

Test Data

Enter a number: 9474



Expected Output

9474 is a Armstrong number



Source Code

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

int main(){
    int n, count = 0, temp, arm = 0, rem;

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

    temp = n;

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

    temp = n;

    while(temp != 0){
        rem = temp % 10;
        arm += pow(rem, count);
        temp /= 10;
    }

    printf("Arm = %d n = %d", arm, n);

    if(arm == n)
        printf("%d is a Armstrong number.", n);
    else
        printf("%d is not an Armstrong number.", n);
}
```

```
    return 0;  
}
```

## 24. WAP. to check whether it is a Palindrome number or not. ↴

Palindrome number

that remains the *same when its digits are reversed.*

Test Data

```
Enter a number: 35153
```



Expected Output

```
35153 is a Palindrome number
```



Source Code

```
#include <stdio.h>  
  
int main(){  
    int num, rev = 0, rem, temp;  
  
    printf("Enter a number: ");  
    scanf("%d", &num);  
  
    temp = num;  
  
    while(temp != 0){  
        rem = temp % 10;  
        rev = rev * 10 + rem;  
        temp /= 10;  
    }  
  
    if(rev == num)  
        printf("%d is a Palindrome number.", num);  
    else  
        printf("%d is not a Palindrome number.", num);  
  
    return 0;  
}
```



## 25. WAP. to print Armstrong number between 1 to n. ↴

Test Data

```
Enter the value of n: 10000
```



### Expected Output

```
1 2 3 4 5 6 7 8 9 153 370 371 407 1634 8208 9474
```



### Source Code

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

int main(){
    int n, count = 0, temp, rem, arm = 0;

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

    for(int i = 1; i <= n; i++){
        arm = 0;
        count = 0;
        temp = i;
        while(temp != 0){
            count++;
            temp /= 10;
        }
        temp = i;
        while(temp != 0){
            rem = temp % 10;
            arm += pow(rem, count);
            temp /= 10;
        }

        if(i == arm)
            printf("%d ", i);
    }
    return 0;
}
```



## 26. WAP. to print n fibonacci series. ↗

### Fibonacci numbers

The Fibonacci numbers are the numbers in the following integer sequence: **0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, .....**

### Test Data

```
Enter the value of n: 5
```



### Expected Output

```
0 1 1 2 3
```



### Source Code

```
#include <stdio.h>

int main(){
    int n, n1 = 0, n2 = 1, sum = 0;

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

    for(int i = 0; i <= n; i++){
        if(i <= 1)
            printf("%d ", i);
        else{
            sum = n1 + n2;
            n1 = n2;
            n2 = sum;
            printf("%d ", sum);
        }
    }

    return 0;
}
```

## 27. WAP. to print n Square number series ex:- 1, 4, 9, 16, 25, 36, 48; ↗

Test Data

```
Enter the value of n: 6
```

Expected Output

```
1, 4, 9, 16, 25, 36
```

Source Code

```
#include <stdio.h>

int main(){
    int n;

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

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

    return 0;
}
```

## 28. WAP. to print the following series 1, 8, 27, 64, 125 , 216.... ↗

#### Test Data

Enter the value of n: 5



#### Expected Output

1 8 27 64 125



#### Source Code

```
#include <stdio.h>

int main(){
    int n;

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

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

    return 0;
}
```



## 29. WAP. to print the following 1, -1, 1, -1, 1, ..... ↴

#### Test Data

Enter the value of n: 5



#### Expected Output

1 -1 1 -1 1



#### Source Code

```
#include <stdio.h>

int main(){
    int n, num = 1;

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

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



```
    return 0;  
}
```

### 30. WAP. to print the following 1, 1, 2, 4, 7, 13, 24, ... (Lucas series) ↗

Test Data

Enter the value of n: 5



Expected Output

1 1 2 4 7



Source Code

```
#include <stdio.h>  
  
int main(){  
    int n, n1 = 1, n2 = 1, n3 = 0, temp;  
  
    printf("Enter the value of n: ");  
    scanf("%d", &n);  
  
    for(int i = 1; i <= n; i++){  
        if(i <= 2){  
            printf("%d ", 1);  
        } else {  
            temp = n1 + n2 + n3;  
            n3 = n2;  
            n2 = n1;  
            n1 = temp;  
  
            printf("%d ", n1);  
        }  
    }  
    return 0;  
}
```

### 31. WAP. to print 1 to n Triangular number sequence. ↗

Triangular number

1, 3, 6, 10, 15, 21, 28, 36, 45,...



Test Data

Enter the value of n: 10



Expected Output

1 3 6 10 15 21 28 36 45 55



### Source Code

```
#include <stdio.h>

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

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

    for(int i = 1; i <= n; i++){
        sum += i;
        printf("%d ", sum);
    }

    return 0;
}
```



## 32. WAP. to print the following sequence 1, 2, 0, 3, -1, 4, -2, 5, -3, 6, -4, 7, .....n ↗

### Test Data

Enter the value of n: 10



### Expected Output

1 2 0 3 -1 4 -2 5 -3 6



### Source Code

```
#include <stdio.h>

int main(){
    int n, n1 = 2, n2 = 1;

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

    for(int i = 1; i <= n; i++){
        if(i % 2 == 0)
            printf("%d ", n1++);
        else
            printf("%d ", n2--);
    }

    return 0;
}
```



### 33. WAP. to print the following sequence 1, 5, 2, 4, 3, 3, 4, 2, 5, 1 ↗

Test Data

Enter the value of n: 20



Expected Output

1 10 2 9 3 8 4 7 5 6 6 5 7 4 8 3 9 2 10 1



Source Code

```
#include <stdio.h>

int main(){
    int n, n1 = 1, n2;

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

    n2 = n / 2;

    for(int i = 1; i <= n; i++){
        if(i % 2 == 0)
            printf("%d ", n2--);
        else
            printf("%d ", n1++);
    }

    return 0;
}
```

### 34. WAP. to print the following sequence 1, 5, 2, 6, 3, 7, 4, 8, 5, 9 ↗

Test Data

Enter the value of n: 20



Expected Output

1 10 2 11 3 12 4 13 5 14 6 15 7 16 8 17 9 18 10 19



Source Code

```
#include <stdio.h>

int main(){
    int n, n1, n2;

    printf("Enter the value of n: ");
    scanf("%d", &n);
```

```
n1 = 1;
n2 = n / 2;

for(int i = 1; i <= n; i++){
    if(i % 2 == 0)
        printf("%d ", n2++);
    else
        printf("%d ", n1++);
}

return 0;
}
```

### 35. WAP. to input a number and print sum of its even and odd digits. ↗

Test Data

Enter any number: 3875698



Expected Output

Sum of even number = 22 and odd number = 24



Source Code

```
#include <stdio.h>

int main(){
    int evenSum = 0, oddSum = 0, num, rem;

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

    while(num != 0){
        rem = num % 10;
        rem % 2 == 0 ? evenSum += rem : (oddSum += rem);
        num /= 10;
    }

    printf("Sum of even number = %d and odd number = %d", evenSum, oddSum);

    return 0;
}
```

### 36. WAP. to input any number and print its factors. ↗

Test Data

Enter any number: 36



### Expected Output

```
1 2 3 4 6 9 12 18 36
```



### Source Code

```
#include <stdio.h>

int main(){
    int num;

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

    for(int i = 1; i <= num; i++){
        if(num % i == 0)
            printf("%d ", i);
    }
    return 0;
}
```



## 37. WAP. to input a number and check it for Strong number. ↗

### Strong number

Strong number is a special number *whose sum of the factorial of digits is equal to the original number.*  
For Example: **145 is strong number.** Since,  $1! + 4! + 5! = 145$ .

### Test Data

```
Enter any number: 40585
```



### Expected Output

```
40585 is a strong number
```



### Source Code

```
#include <stdio.h>

int main(){
    int num, sum = 0, temp, fact = 1, rem;

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

    temp = num;

    while(temp != 0){
        fact = 1;
        rem = temp % 10;
        for(int i = rem; i >= 1; i--){
            fact *= i;
        }
        sum += fact;
        temp /= 10;
    }
    if(sum == num)
        printf("%d is a strong number", num);
    else
        printf("%d is not a strong number", num);
}
```



```

        fact *= i;
    }
    sum += fact;
    temp /= 10;
}

if(sum == num)
    printf("%d is a strong number", num);
else
    printf("%d is not a strong number", num);
return 0;
}

```

### 38. WAP. to input n number and print all strong numbers in between 1 to n. ↗

#### Strong number

Strong number is a special number *whose sum of the factorial of digits is equal to the original number.*  
For Example: **145 is strong number.** Since,  $1! + 4! + 5! = 145$ .

#### Test Data

Enter the value of n: 50000



#### Expected Output

1 2 145 40585



#### Source Code

```

#include <stdio.h>

int main(){
    int n, temp, fact, sum, rem;

    printf("\nEnter the value of n: ");
    scanf("%d", &n);

    for(int i = 1; i <= n; i++){
        temp = i;
        sum = 0;
        while(temp != 0){
            fact = 1;
            rem = temp % 10;
            for(int j = rem; j >= 1; j--)
                fact *= j;
            sum += fact;
            temp /= 10;
        }

        if(sum == i)
            printf("%d ", i);
    }
}

```

```
    return 0;  
}
```

### 39. WAP. to input a number and print the min and max digit of the number. ↗

Test Data

Enter a number: 312



Expected Output

Min = 1 Max = 3



Source Code

```
#include <stdio.h>  
  
int main(){  
    int num, min = 9, max = 0, rem;  
  
    printf("Enter a number: ");  
    scanf("%d", &num);  
  
    while(num != 0){  
        rem = num % 10;  
        if(rem > max)  
            max = rem;  
        if(rem < min)  
            min = rem;  
        num /= 10;  
    }  
  
    printf("Min = %d Max = %d", min, max);  
    return 0;  
}
```

### 40. WAP. to input a number and make a new number by adding 1 with individual digit. ↗

Test Data

Enter a number: 239



Expected Output

Sum = 350



Source Code

```
#include <stdio.h>

int main(){
    int num, count = 0, sum, temp;

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

    temp = num;

    while(temp != 0){
        count = count * 10 + 1;
        temp /= 10;
    }

    sum = num + count;

    printf("Sum = %d", sum);
    return 0;
}
```

## 41. WAP. to check perfect number or not. ↗

### Perfect number

A perfect number is a positive integer that is *equal to the sum of its factors except for the number itself.*

*The smallest perfect number is 6, which is the sum of its factors: 1, 2, and 3.*

Note that this sum does not include the number itself which is also a factor of itself.

### Test Data

```
Enter a number: 496
```

### Expected Output

```
Perfect number
```

### Source Code

```
#include <stdio.h>

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

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

    for(int i = 1; i <= num / 2; i++){
        if(num % i == 0)
            sum += i;
    }
}
```

```
    if(sum == num)
        printf("Perfect number");
    else
        printf("Not a Perfect number");

    return 0;
}
```

## 42. WAP. to print all integers that are not divisible by either 2 or 3 and lie between 1 and n. ↗

Test Data

Enter the value of n: 20



Expected Output

1 5 7 11 13 17 19



Source Code

```
#include <stdio.h>

int main(){
    int n;

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

    for(int i = 1; i <= n; i++){
        if(i % 2 != 0 && i % 3 != 0)
            printf("%d ", i);
    }
    return 0;
}
```

## 43. WAP. to input a number and print the position of individual digit. ↗

Test Data

Enter a number: 123



Expected Output

100 20 3



Source Code

```

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

int main(){
    int count = 0, rem, num, t;

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

    t = num;

    while(t){
        count++;
        t /= 10;
    }

    for(int i = count - 1; i >= 0; i--){
        rem = num / pow(10, i);
        rem = rem * pow(10, i);
        printf("%d ", rem);
        num = num % (int)pow(10, i);
    }

    return 0;
}

```

#### 44. WAP. to express a given number as a sum of two prime numbers. Print all possible solutions. ↴

Test Data

Enter a number: 34

Expected Output

3 + 31 = 34  
 5 + 29 = 34  
 11 + 23 = 34  
 15 + 19 = 34  
 17 + 17 = 34

Source Code

```

#include <stdio.h>

int main(){
    int num, c, c1, subVal;

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

    for(int i = 2; i <= num / 2; i++){
        c = 1;
        if(i == 2){

```

```

        c = 1;
    } else if(i % 2 == 0){
        c = 0;
    }

    if(c == 1){
        subVal = num - i;
        c1 = 1;
        for(int j = 2; j <= subVal/2; j++){
            if(subVal % j == 0){
                c1 = 0;
                break;
            }
        }
        if(c1 == 1){
            printf("\n%d + %d = %d", i, subVal, i + subVal);
        }
    }
    if(c1 != 1){
        printf("Sum of Prime possible number is 0");
    }
    return 0;
}

```

## 45. WAP. to input a number and check it is square of 2 or not ↗

Test Data

Enter any number: 64



Expected Output

64 is a square of 2



Source Code

```

#include <stdio.h>

int main(){
    int n, s = 1;

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

    for(int i = 1; i <= n; i++){
        s *= 2;
        if(s == n){
            break;
        }
    }

    if(s == n){
        printf("%d is a square of 2", n);
    } else{

```

```
    printf("%d is not a square of 2", n);
}

return 0;
}
```

## 46. WAP. to swap first and last digit of a number ↗

### Test Data

Enter any number: 452



### Expected Output

252



### Source Code

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

int main(){
    int num, temp, count = 0, swappedNum, firstDigit, lastDigit;

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

    temp = num;

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

    lastDigit = num % 10;
    firstDigit = num / (int)pow(10, count - 1);
    swappedNum = num % (int)pow(10, count - 1);
    swappedNum -= lastDigit;
    swappedNum += firstDigit;
    swappedNum += lastDigit * (int)pow(10, count - 1);

    printf("%d", swappedNum);
    return 0;
}
```

## 47. WAP. to find all prime factors of a number. ↗

### Test Data

Enter any number: 88



### Expected Output

2 11



### Source Code

```
#include <stdio.h>

int main(){
    int num, count;

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

    for(int i = 1; i <= num; i++){
        count = 0;
        if(num % i == 0){
            for(int j = 1; j <= i; j++){
                if(i % j == 0){
                    count++;
                }
            }
        }

        if(count == 2){
            printf("%d ", i);
        }
    }
    return 0;
}
```



## 48. WAP. to convert Decimal to Binary number system. ↗

### Test Data

Enter any number: 10



### Expected Output

Decimal number = 2



### Source Code

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

int main(){
    int dec = 0, i = 0, rem;
    long long bin;
    printf("Enter a number: ");
    scanf("%lld", &bin);

    while(bin != 0){
```



```
    rem = bin % 10;
    dec += rem * (int)pow(2, i);
    i++;
    bin /= 10;
}

printf("Decimal number = %d", dec);
return 0;
}
```

## 49. WAP. to convert Octal to Decimal number system. ↗

Test Data

Enter any number: 16



Expected Output

Decimal number = 14



Source Code

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

int main(){
    int octal, dec = 0, i = 0, rem;

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

    while(octal != 0){
        rem = octal % 10;
        dec += rem * (int)pow(8, i);
        i++;
        octal /= 10;
    }

    printf("Decimal number = %d", dec);
    return 0;
}
```

## 50. WAP. to convert Decimal to Binary number system. ↗

Test Data

Enter any number: 10



Expected Output

Binary number = 1010



#### Source Code

```
#include <stdio.h>

int main(){
    int dec, rem, i = 1;
    long long bin = 0;

    printf("Enter any number: ");
    scanf("%d", &dec);

    while(dec != 0){
        rem = dec % 2;
        bin += rem * i;
        i *= 10;
        dec /= 2;
    }

    printf("Binary number = %d", bin);
    return 0;
}
```



## 51. WAP. to convert Decimal to Octal number system. ↗

#### Test Data

Enter any number: 14



#### Expected Output

Octal number = 16



#### Source Code

```
#include <stdio.h>

int main(){
    int dec, rem, i = 1, octal = 0;

    printf("Enter any number: ");
    scanf("%d", &dec);

    while(dec != 0){
        rem = dec % 8;
        octal += rem * i;
        i *= 10;
        dec /= 8;
    }

    printf("Octal number = %d", octal);
```



```
    return 0;  
}
```

## Patterns ↗

### 1. WAP. to print the following pattern ↗

Test Data

```
Enter the value of n: 5
```



Expected Output

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



Source Code

```
# include <stdio.h>  
  
int main(){  
    int n;  
  
    printf("Enter the value of n: ");  
    scanf("%d", &n);  
  
    for(int i = 1; i <= n; i++){  
        for(int j = 1; j <= n; j++){  
            printf("*");  
        }  
        printf("\n");  
    }  
  
    return 0;  
}
```

### 2. WAP. to print the following pattern ↗

Test Data

```
Enter the value of n: 5
```



Expected Output

```
*
```

```
* *
```

```
* * *
```

```
* * * *
```

```
* * * * *
```



## Source Code

```
#include <stdio.h>

int main(){
    int n;

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

    for(int i = 1; i <= n; i++){
        for(int j = 1; j <= i; j++){
            printf("*");
        }
        printf("\n");
    }

    return 0;
}
```



## 3. WAP. to print the following pattern ↴

### Test Data

```
Enter the value of n: 5
```



### Expected Output

```
*
```

```
* *
```

```
* * *
```

```
* * *
```

```
*
```



## Source Code

```
#include <stdio.h>

int main(){
    int n;

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

    for(int i = n; i >= 1; i--){
        for(int j = 1; j <= i; j++){
            printf("*");
        }
        printf("\n");
    }

    return 0;
}
```



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

    return 0;
}
```

#### 4. WAP. to print the following pattern ↗

Test Data

Enter the value of n: 4



Expected Output

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



Source Code

```
#include <stdio.h>

int main(){
    int n;

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

    for(int i = 1; i <= n; i++){
        for(int j = n; j >= i; j--){
            printf(" ");
        }
        for(int j = 1; j <= i; j++){
            printf("*");
        }
        printf("\n");
    }

    return 0;
}
```

#### 5. WAP. to print the following pattern ↗

Test Data

Enter the value of n: 4



Expected Output

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



#### Source Code

```
#include <stdio.h>

int main(){
    int n;

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

    for(int i = 1; i <= n; i++){
        for(int j = 1; j <= i; j++){
            printf(" ");
        }
        for(int j = n; j >= i; j--){
            printf("*");
        }
        printf("\n");
    }

    return 0;
}
```



#### 6. WAP. to print the following pattern ↴

##### Test Data

```
Enter the value of n: 4
```



##### Expected Output

```
*
```

```
* * *
```

```
* * * * *
```

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



#### Source Code

```
#include <stdio.h>

int main(){
    int n;

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

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

```



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

return 0;
}
```

## 7. WAP. to print the following pattern ↗

### Test Data

Enter the value of n: 5



### Expected Output

```
*
```

```
* *
```

```
* * *
```

```
* * * *
```

```
* * * * *
```

```
* * * *
```

```
* * *
```

```
* *
```

```
*
```



### Source Code

```
#include <stdio.h>

int main(){
    int n;

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

    for(int i = 1; i <= n; i++){
        for(int j = 1; j <= i; j++){
            printf("* ");
        }
        printf("\n");
    }
    for(int i = 1; i <= n - 1; i++){
        for(int j = 1; j <= n - i; j++){
            printf("* ");
        }
        printf("\n");
    }

    return 0;
}
```

## 8. WAP. to print the following pattern ↴

Test Data

```
Enter the value of n: 5
```



Expected Output

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



Source Code

```
#include <stdio.h>  
  
int main(){  
    int n;  
  
    printf("Enter the value of n: ");  
    scanf("%d", &n);  
  
    for(int i = 1; i <= n; i++){  
        for(int j = 1; j <= n - i; j++){  
            printf(" ");  
        }  
        for(int j = 1; j <= i; j++){  
            printf("* ");  
        }  
        printf("\n");  
    }  
  
    for(int i = 1; i <= n - 1; i++){  
        for(int j = 1; j <= i; j++){  
            printf(" ");  
        }  
        for(int j = n - i; j >= 1; j--){  
            printf("* ");  
        }  
        printf("\n");  
    }  
  
    return 0;  
}
```



## 9. WAP. to print the following pattern ↴

Test Data

Enter the value of n: 5



#### Expected Output

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



#### Source Code

```
#include <stdio.h>  
  
int main(){  
    int n;  
  
    printf("Enter the value of n: ");  
    scanf("%d", &n);  
  
    for(int i = 1; i <= n; i++){  
        for(int j = 1; j <= n - i; j++){  
            printf(" ");  
        }  
        for(int j = 1; j <= (i * 2) - 1; j++){  
            printf("*");  
        }  
        printf("\n");  
    }  
  
    for(int i = n - 1; i >= 1; i--){  
        for(int j = 1; j <= n - i; j++){  
            printf(" ");  
        }  
        for(int j = 1; j <= (i * 2) - 1; j++){  
            printf("*");  
        }  
        printf("\n");  
    }  
  
    return 0;  
}
```



## 10. WAP. to print the following pattern ☺

#### Test Data

Enter the value of n: 5



### Expected Output

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



### Source Code

```
#include <stdio.h>  
  
int main(){  
    int n;  
  
    printf("Enter the value of n: ");  
    scanf("%d", &n);  
  
    for(int i = 1; i <= n; i++){  
        for(int j = 1; j <= 2 * (n - i) + 1; j++){  
            printf("* ");  
        }  
        printf("\n");  
    }  
  
    for(int i = 1; i <= n - 1; i++){  
        for(int j = 1; j <= (2 * i) + 1; j++){  
            printf("* ");  
        }  
        printf("\n");  
    }  
  
    return 0;  
}
```



## 11. WAP. to print the following pattern ↴

### Test Data

```
Enter the value of n: 5
```



### Expected Output

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



```
* * * * *
```

## Source Code

```
#include <stdio.h>

int main(){
    int n;

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

    for(int i = 1; i <= n; i++){
        for(int j = 1; j <= (i - 1) * 2; j++){
            printf(" ");
        }
        for(int j = 1; j <= 2 * (n - i) + 1; j++){
            printf("* ");
        }
        printf("\n");
    }
    for(int i = 1; i <= n - 1; i++){
        for(int j = 1; j <= 2 * (n - i) - 2; j++){
            printf(" ");
        }
        for(int j = 1; j <= (2 * i) + 1; j++){
            printf("* ");
        }
        printf("\n");
    }

    return 0;
}
```



## 12. WAP. to print the following pattern ↴

### Test Data

```
Enter the value of n: 5
```



### Expected Output

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



## Source Code

```
#include <stdio.h>

int main(){
    int n;

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

    for(int i = 1; i <= n; i++){
        for(int j = 1; j <= i; j++){
            printf(" ");
        }
        for(int j = 1; j <= 2 * (n - i) + 1; j++){
            printf("* ");
        }
        printf("\n");
    }

    for(int i = 1; i <= n - 1; i++){
        for(int j = 1; j <= n - i; j++){
            printf(" ");
        }
        for(int j = 1; j <= (2 * i) + 1; j++){
            printf("* ");
        }
        printf("\n");
    }
    return 0;
}
```

### 13. WAP. to print the following pattern ↗

Test Data

```
Enter the value of n: 5
```

Expected Output

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

Source Code

```
#include <stdio.h>

int main(){
    int n;

    printf("Enter the value of n: ");
    scanf("%d", &n);
```

```

for(int i = 1; i <= n; i++){
    for(int j = 1; j <= n; j++){
        if(i == 1 || j == 1){
            printf("*");
        } else if(i == n || j == n){
            printf("*");
        }

        if(i != 1 && j != 1 && i != n && j != n){
            printf(" ");
        }
    }
    printf("\n");
}

return 0;
}

```

## 14. WAP. to print the following pattern ↗

Test Data

Enter the value of n: 5



Expected Output

A  
A B  
A B C  
A B C D  
A B C D E



Source Code

```

#include <stdio.h>

int main(){
    int n, a;

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

    for(int i = 1; i <= n; i++){
        a = 65;
        for(int j = 1; j <= i; j++){
            printf("%c", a++);
        }
        printf("\n");
    }
    return 0;
}

```

## 15. WAP. to print the following pattern ↴

Test Data

Enter the value of n: 5



Expected Output

A  
B C  
D E F  
G H I J  
K L M N O



Source Code

```
#include <stdio.h>

int main(){
    int n, a = 65;

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

    for(int i = 1; i <= n; i++){
        for(int j = 1; j <= i; j++){
            printf("%c", a++);
        }
        printf("\n");
    }
    return 0;
}
```



## 16. WAP. to print the following pattern ↴

Test Data

Enter the value of n: 5



Expected Output

A  
A B  
A B C  
A B C D  
A B C D E



Source Code

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



```
int main(){
    int n, a;

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

    for(int i = 1; i <= n; i++){
        a = 65;
        for(int j = 1; j <= n - i; j++){
            printf(" ");
        }
        for(int j = 1; j <= i; j++){
            printf("%c", a++);
        }
        printf("\n");
    }

    return 0;
}
```

## 17. WAP. to print the following pattern ↗

| Test Data

Enter the value of n: 5



| Expected Output

E  
D E  
C D E  
B C D E  
A B C D E



| Source Code

```
#include <stdio.h>

int main(){
    int n, a;

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

    for(int i = 1; i <= n; i++){
        a = 65;
        for(int j = n - i; j < n; j++){
            printf("%c", a + j);
        }
        printf("\n");
    }

    return 0;
}
```

## 18. WAP. to print the following pattern ↴

Test Data

Enter the value of n: 5



Expected Output

E  
D E  
C D E  
B C D E  
A B C D E



Source Code

```
#include <stdio.h>

int main(){
    int n, a;

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

    for(int i = 1; i <= n; i++){
        a = 65;
        for(int j = n - i; j >= 1; j--){
            printf(" ");
        }
        for(int j = n - i; j < n; j++){
            printf("%c", a + j);
        }
        printf("\n");
    }

    return 0;
}
```



## 19. WAP. to print the following pattern ↴

Test Data

Enter the value of n: 5



Expected Output

A  
B C  
C D E



```
D E F G  
E F G H I
```

#### Source Code

```
#include <stdio.h>

int main(){
    int n, a;

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

    for(int i = 1; i <= n; i++){
        a = 65 + i - 1;
        for(int j = 1; j <= i; j++){
            printf("%c", a++);
        }
        printf("\n");
    }

    return 0;
}
```



#### 20. WAP. to print the following pattern ↴

##### Test Data

```
Enter the value of n: 5
```



##### Expected Output

```
1 2 3 4 5  
2 3 4 5 6  
3 4 5 6 7  
4 5 6 7 8  
5 6 7 8 9
```



#### Source Code

```
#include <stdio.h>

int main(){
    int n;

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

    for(int i = 1; i <= n; i++){
        for(int j = i; j <= n + i - 1; j++){
            printf("%d", j);
        }
        printf("\n");
    }
}
```



```
    return 0;  
}
```

## 21. WAP. to print the following pattern ↗

Test Data

Enter the value of n: 5



Expected Output

```
5 4 3 2 1  
6 5 4 3 2  
7 6 5 4 3  
8 7 6 5 4  
9 8 7 6 5
```



Source Code

```
#include <stdio.h>  
  
int main(){  
    int n;  
  
    printf("Enter the value of n: ");  
    scanf("%d", &n);  
  
    for(int i = 1; i <= 5; i++){  
        for(int j = n + i - 1; j >= i; j--){  
            printf("%2d", j);  
        }  
        printf("\n");  
    }  
    return 0;  
}
```



## 22. WAP. to print the following pattern ↗

Test Data

Enter the value of n: 5



Expected Output

```
1 2 3 4 5  
2 4 6 8 10  
3 6 9 12 15  
4 8 12 16 20  
5 10 15 20 25
```



## Source Code

```
#include <stdio.h>

int main(){
    int n;

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

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

    return 0;
}
```



## 23. WAP. to print the following pattern ↴

### Test Data

```
Enter the value of n: 5
```



### Expected Output

```
1 2 3 4 5
2 3 4 5 6
4 5 6 7 8
7 8 9 10 11
11 12 13 14 15
```



## Source Code

```
#include <stdio.h>

int main(){
    int n, temp, count = 1;

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

    for(int i = 1; i <= n; i++){
        temp = count;
        for(int j = 1; j <= n; j++){
            printf("%d ", count);
            count++;
        }
        count = temp + i;
        printf("\n");
    }
}
```



```
    return 0;
}
```

## 24. WAP. to print the following pattern ↴

Test Data

Enter the value of n: 5



Expected Output

```
1=1
1+2=3
1+2+3=6
1+2+3+4=10
1+2+3+4+5=15
```



Source Code

```
#include <stdio.h>

int main(){
    int n, sum;

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

    for(int i = 1; i <= n; i++){
        sum = 0;
        for(int j = 1; j <= i; j++){
            sum += j;
            printf("%d", j);
            if(j == i)
                printf("=%d", sum);
            else
                printf("+");
        }
        printf("\n");
    }

    return 0;
}
```



## 25. WAP. to print the following pattern ↴

Test Data

Enter the value of n: 5



Expected Output

```
1=1  
1+2=3  
1+2+3=6  
1+2+3+4=10  
1+2+3+4+5=15
```



#### Source Code

```
#include <stdio.h>

int main(){
    int n, sum;

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

    for(int i = 1; i <= n; i++){
        sum = 0;
        for(int j = 1; j <= n - i; j++){
            printf(" ");
        }
        for(int j = 1; j <= i; j++){
            sum += j;
            printf("%d", j);
            if(j == i)
                printf("=%d", sum);
            else
                printf("+");
        }
        printf("\n");
    }

    return 0;
}
```



## 26. WAP to print the following pattern ↴

#### Test Data

```
Enter the value of n: 5
```



#### Expected Output

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



#### Source Code

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



```
int main(){
    int n;

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

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

    return 0;
}
```

## 27. WAP. to print the following pattern ↴

Test Data

```
Enter the value of n: 5
```



Expected Output

```
1
2 3
3 4 5
4 5 6 7
5 6 7 8 9
```



Source Code

```
#include <stdio.h>

int main(){
    int n, count = 1;

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

    for(int i = 1; i <= n; i++){
        count = i;
        for(int j = 1; j <= i; j++){
            printf("%2d", count++);
        }
        printf("\n");
    }

    return 0;
}
```

## 28. WAP. to print the following pattern ↴

#### Test Data

Enter the value of n: 5



#### Expected Output

1  
2 1  
1 2 1  
2 1 2 1  
1 2 1 2 1



#### Source Code

```
#include <stdio.h>

int main(){
    int n, count;

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

    for(int i = 0; i < n; i++){
        for(int j = 0; j <= i; j++){
            if((i + j) % 2 == 0)
                printf("1 ");
            else
                printf("2 ");
        }
        printf("\n");
    }

    return 0;
}
```



## 29. WAP. to print the following pattern ↴

#### Test Data

Enter the value of n: 5



#### Expected Output

1  
0 1  
1 0 1  
0 1 0 1  
1 0 1 0 1



#### Source Code

```
#include <stdio.h>

int main(){
    int n;

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

    for(int i = 1; i <= n; i++){
        for(int j = i; j >= 1; j--){
            printf("%2d", j%2);
        }
        printf("\n");
    }

    return 0;
}
```

### 30. WAP. to print the following pattern ↴

Test Data

```
Enter the value of n:
```

Expected Output

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

Source Code

```
#include <stdio.h>

int main(){
    int n;

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

    for(int i = 1; i <= n; i++){
        for(int j = 1; j <= i; j++){
            printf("%2d", j%2);
        }
        printf("\n");
    }

    return 0;
}
```

### 31. WAP. to print the following pattern ↴

Test Data

Enter the value of n: 5



Expected Output

1  
2 3  
4 5 6  
7 8 9 10  
11 12 13 14 15



Source Code

```
#include <stdio.h>

int main(){
    int n, count = 1;

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

    for(int i = 1; i <= n; i++){
        for(int j = 1; j <= i; j++){
            printf("%2d ", count++);
        }
        printf("\n");
    }

    return 0;
}
```



### 32. WAP. to print the following pattern ↴

Test Data

Enter the value of n: 5



Expected Output

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



Source Code

```
#include <stdio.h>

int main(){
    int n;

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

    for(int i = 1; i <= n; i++){
        for(int j = n - (i - 1); j <= n; j++){
            printf("%d ", j);
        }
        printf("\n");
    }

    return 0;
}
```

### 33. WAP. to print the following pattern ↴

Test Data

```
Enter the value of n: 5
```

Expected Output

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

Source Code

```
#include <stdio.h>

int main(){
    int n;

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

    for(int i = 1; i <= n; i++){
        for(int j = n - (i - 1); j >= n; j--){
            printf("%d ", j);
        }
        printf("\n");
    }

    return 0;
}
```

### 34. WAP. to print the following pattern ↴

Test Data

Enter the value of n: 5



Expected Output

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



Source Code

```
#include <stdio.h>

int main(){
    int n;

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

    for(int i = 1; i <= n; i++){
        for(int j = i; j >= 1; j--){
            printf("%d", j);
        }
        printf("\n");
    }

    return 0;
}
```



### 35. WAP. to print the following pattern ↴

Test Data

Enter the value of n: 5



Expected Output

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



Source Code

```
#include <stdio.h>

int main(){
    int n;

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

    for(int i = 1; i <= n; i++){
        for(int j = 1; j <= n - i; j++){
            printf(" ");
        }
        for(int j = i; j >= 1; j--){
            printf("%d ", j);
        }
        printf("\n");
    }

    return 0;
}
```

### 36. WAP. to print the following pattern ↴

Test Data

```
Enter the value of n: 5
```

Expected Output

```

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

Source Code

```
#include <stdio.h>

int main(){
    int n;

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

    for(int i = 1; i <= n; i++){
        for(int j = 1; j <= n - i; j++){
            printf(" ");
        }
        for(int j = 1; j <= i + (i - 1); j++){
            printf("*");
        }
        printf("\n");
    }

}
```

```
    return 0;
}
```

### 37. WAP. to print the following pattern ↴

Test Data

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



Expected Output

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



Source Code

```
#include <stdio.h>

int main(){
    int n;

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

    for(int i = 1; i <= n; i++){
        for(int j = 1; j <= n - i + 1; j++){
            printf(" ");
        }
        for(int j = 1; j <= i + (i - 1); j++){
            printf("*");
        }
        printf("\n");
    }
    for(int i = 1; i <= n; i++){
        for(int j = 1; j <= i; j++){
            printf(" ");
        }
        for(int j = 1; j <= 2 * (n - i) + 1; j++){
            printf("*");
        }
        printf("\n");
    }
    return 0;
}
```



### 38. WAP. to print the following pattern ↴

### Test Data

Enter the value of n: 5



### Expected Output

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



### Source Code

```
#include <stdio.h>

int main(){
    int n;

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

    for(int i = 1; i <= n; i++){
        for(int j = 1; j <= n - i; j++){
            printf(" ");
        }
        for(int j = 1; j <= i; j++){
            printf("%d", j);
        }
        for(int j = i - 1; j >= 1; j--){
            printf("%d", j);
        }
        printf("\n");
    }

    return 0;
}
```



## 39. WAP. to print the following pattern ↴

### Test Data

Enter the value of n: 5



### Expected Output

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



### Source Code

```
#include <stdio.h>

int main(){
    int n;

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

    for(int i = 1; i <= n; i++){
        for(int j = 1; j <= n - i + 1; j++){
            printf(" ");
        }
        for(int j = n; j >= n - i + 1; j--){
            printf("%d ", j);
        }
        for(int j = n - i + 1; j < n; j++){
            printf("%d ", j + 1);
        }
        printf("\n");
    }

    return 0;
}
```



### 40. WAP. to print the following pattern ↴

#### Test Data

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



#### Expected Output

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



### Source Code

```
#include <stdio.h>

int main(){
    int n;

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

    for(int i = 1; i <= n; i++){
        for(int j = 1; j <= n - i + 1; j++){
            printf(" ");
        }
        for(int j = n; j >= n - i + 1; j--){
            printf("%d ", j);
        }
        for(int j = n - i + 1; j < n; j++){
            printf("%d ", j + 1);
        }
        printf("\n");
    }

    return 0;
}
```



```

    }
    for(int j = 1; j <= i; j++){
        printf("%d ", j);
    }
    for(int j = i - 1; j >= 1; j--){
        printf("%d ", j);
    }
    printf("\n");
}

for(int i = n; i >= 1; i--){
    for(int j = i; j <= n; j++){
        printf(" ");
    }
    for(int j = 1; j <= i; j++){
        printf("%d ", j);
    }
    for(int j = i - 1; j >= 1; j--){
        printf("%d ", j);
    }
    printf("\n");
}
return 0;
}

```

## 41. WAP. to print the following pattern ↗

### Test Data

Enter the value of n: 5



### Expected Output

```

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

```



### Source Code

```

#include <stdio.h>

int main(){
    int n, c;

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

    c = n;
    for(int i = 1; i <= n; i++){
        if(i == n){
            c = n;
        }
        for(int j = 1; j <= n; j++){

```

```

    if(i == 1){
        printf("%d ", j);
    } else if(j == 1){
        printf("%d ", i);
    } else if(j == n || i == n){
        printf("%d ", --c);
    } else {
        printf(" ");
    }

}
printf("\n");
}

return 0;
}

```

## Switch Case Statement ↗

### 1. Compute the following: ↗

1. Factorial of a number
2. Prime or not
3. Odd or even
4. Exit

Once a menu item is selected the appropriate action should be taken and once this action is finished, the menu should reappear. Unless the user selects the 'Exit' option the program should continue to run.

**Hint:** Make use of an infinite while and a switch statement.

#### Test Data

```

Enter a character: (Factorial 'f' Prime 'p' Even or Odd 'e' or Exit ''): f
□

Enter a number: 5
Factorial = 120
Enter a character: (Factorial 'f' Prime 'p' Even or Odd 'e' or Exit ''): p
□

Enter a number: 15
Not a Prime Number
Enter a character: (Factorial 'f' Prime 'p' Even or Odd 'e' or Exit ''): t
□

```

#### Source Code

```

#include <stdio.h>

int main(){
    char ch;

```

```

int a = 1, flag = 1, num, fact = 1;

while(a){
    printf("\nEnter a character:
(Factorial 'f' Prime 'p' Even or Odd 'e' or Exit ''): ");
    scanf(" %c", &ch);

    switch(ch){
        case 'f':
            printf("\nEnter a number: ");
            scanf("%d", &num);

            for(int i = num; i >= 1; i--){
                fact *= i;
            }

            printf("Factorial = %d", fact);
            break;
        case 'p':
            printf("\nEnter a number: ");
            scanf("%d", &num);

            for(int i = 2; i <= num / 2; i++){
                flag = 1;
                if(num % i == 0){
                    flag = 0;
                    break;
                }
            }

            if(flag == 1){
                printf("\nPrime Number");
            } else {
                printf("\nNot a Prime Number");
            }
            break;
        case 'e':
            printf("\nEnter a number: ");
            scanf("%d", &num);

            if(num % 2 == 0){
                printf("\nEven Number");
            } else{
                printf("\nOdd Number");
            }
            break;
        default:
            a = 0;
    }
}

return 0;
}

```

## Functions ↗

---

## 1. WAP. To Accept Two Numbers From User And Print It's Addition,Subtraction,Multiplication,Division Using Different Functions In C Language ↗

Test Data

Enter 2 numbers: 10 5



Expected Output

Sum = 15 Sub = 5 Mul = 50 Div = 2



Source Code

```
#include <stdio.h>

int addition(int, int);
int subtraction(int, int);
int multiplication(int, int);
int division(int, int);

int main(){
    int x, y, sum, sub, div, mul;

    printf("Enter two numbers: ");
    scanf("%d %d", &x, &y);

    sum = addition(x, y);
    sub = subtraction(x, y);
    mul = multiplication(x, y);
    div = division(x, y);

    printf("Sum = %d Sub = %d Mul = %d Div = %d", sum, sub, mul, div);

    return 0;
}

int addition(int x, int y){
    return x + y;
}

int subtraction(int x, int y){
    return x - y;
}

int multiplication(int x, int y){
    return x * y;
}

int division(int x, int y){
    return x / y;
}
```

## 2. WAP. to print greatest among the two numbers. ↗

#### Test Data

Enter 2 Numbers: 15 20



#### Expected Output

Greater number = 20



#### Source Code

```
#include <stdio.h>

int greater(int, int);

int main(){
    int x, y, max;

    printf("Enter two numbers: ");
    scanf("%d%d", &x, &y);

    max = greater(x, y);

    printf("Greater number = %d", max);

    return 0;
}

int greater(int x, int y){
    if(x > y){
        return x;
    } else {
        return y;
    }
}
```



### 3. Write a function to calculate factorial of a number.(Takes Something, Returns Something) ↴

#### Test Data

Enter a number: 6



#### Expected Output

Factorial = 720



#### Source Code

```
#include <stdio.h>

int factorial(int);
```



```

int main(){
    int n, fact;

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

    fact = factorial(n);

    printf("Factorial = %d", fact);
    return 0;
}

int factorial(int n){
    int f = 1;
    for(int i = n; i >= 1; i--){
        f *= i;
    }

    return f;
}

```

#### 4. Write a function to calculate area of a circle.(Takes Something, Returns Something) ↗

Test Data

Enter radius: 27



Expected Output

Area of a circle = 2289.000000



Source Code

```

#include <stdio.h>

int areaOfCircle(float);

int main(){
    float r, area;

    printf("Enter radius: ");
    scanf("%f", &r);

    area = areaOfCircle(r);

    printf("Area of a Circle = %f", area);
    return 0;
}

int areaOfCircle(float r){
    float pi = 3.141;

```

```
    return pi * r * r;
}
```

## 5. Write a function to calculate sum of first N natural numbers. (all four ways: TNRN, TSRN, TNRS, TSRS) ↴

### Test Data

Enter the value of n: 10



### Expected Output

Sum = 55



### Source Code

```
#include <stdio.h>

int sumOfNaturalNumber(int);

int main(){
    int n, sum;

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

    sum = sumOfNaturalNumber(n);

    printf("Sum = %d", sum);

    return 0;
}

int sumOfNaturalNumber(int n){
    int sum = 0;
    for(int i = 1; i <= n; i++){
        sum += i;
    }
    return sum;
}
```

## 6. Write a function to calculate sum of squares of first N natural numbers.(all four ways: TNRN, TSRN, TNRS, TSRS) ↴

### Test Data

Enter the value of n: 5



### Expected Output

Sum of square of natural = 55



#### Source Code

```
#include <stdio.h>

int sumOfSquareOfNatural(int);

int main(){
    int n, sum;

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

    sum = sumOfSquareOfNatural(n);

    printf("Sum of square of natural = %d", sum);

    return 0;
}

int sumOfSquareOfNatural(int n){
    int sum = 0;

    for(int i = 1; i <= n; i++){
        sum += i * i;
    }

    return sum;
}
```



7. Write a function to express a given number as a sum of two prime numbers.  
Print all possible solutions ↗

#### Test Data

Enter a number: 34



#### Expected Output

3 + 31 = 34  
5 + 29 = 34  
11 + 23 = 34  
17 + 17 = 34



#### Source Code

```
#include <stdio.h>

int isPrime(int);

int main(){
    int num, subVal, flag;
```



```

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

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

    if(isPrime(i)){
        subVal = num - i;
        if(isPrime(subVal)){
            printf("\n%d + %d = %d", i, subVal, i + subVal);
        } else{
            flag = 0;
        }
    }
}

if(flag == 0){
    printf("Sum of prime possible number is 0");
}

return 0;
}

int isPrime(int n){
    for(int i = 2; i <= n/2; i++){
        if(n % i == 0){
            return 0;
        }
    }
    return 1;
}

```

## Recursion ↗

---

### 1. Addition of two numbers using recursion ↗

Test Data

Enter two numbers: 10 5



Expected Output

Sum = 15



Source Code

```

#include <stdio.h>

int main(){
    int sum(int, int);

```



```

int n1, n2, s;

printf("Enter two numbers: ");
scanf("%d%d", &n1, &n2);

s = sum(n1, n2);

printf("Sum = %d", s);
return 0;
}

// Function Defination
int sum(int n1, int n2){
    if(n2 == 0){      // Base Case
        return n1;
    }
    sum(n1 + 1, n2 - 1); // Recursive Case
}

```

## 2. Subtraction of two numbers using recursion. ↗

Test Data

Enter two numbers: 20 5



Expected Output

Sum = 15



Source Code

```

#include <stdio.h>

int main(){
    int sub(int, int);
    int n1, n2, s;

    printf("Enter two numbers: ");
    scanf("%d%d", &n1, &n2);

    s = sub(n1, n2);

    printf("Subtraction = %d", s);
    return 0;
}

int sub(int n1, int n2){
    if(n2 == 0){
        return n1;
    }
    sub(n1 - 1, n2 - 1);
}

```

### 3. Multiplication of two numbers using recursion. ↗

Test Data

Enter two numbers: 20 5



Expected Output

Sum = 100



Source Code

```
#include <stdio.h>

int main(){
    int mul(int, int);
    int n1, n2, m;

    printf("Enter two numbers: ");
    scanf("%d%d", &n1, &n2);

    m = mul(n1, n2);

    printf("Multiplication = %d", m);
    return 0;
}

int mul(int n1, int n2){
    if(n2 == 1){
        return n1;
    }
    return n1 + mul(n1, n2 - 1);
}
```



### 4. Division of two numbers using recursion. ↗

Test Data

Enter two numbers: 20 5



Expected Output

Sum = 4



Source Code

```
#include <stdio.h>

int main(){
    int div(int, int);
    int n1, n2, d;
```



```
printf("Enter two numbers: ");
scanf("%d%d", &n1, &n2);

d = div(n1, n2);

printf("Division = %d", d);
return 0;
}
int div(int n1, int n2){
    if(n1 == 0){
        return 0;
    }
    return 1 + div(n1 - n2, n2);
}
```

## 5. print n to 1 using recursion ↗

Test Data

Enter the value of n: 5



Expected Output

5 4 3 2 1



Source Code

```
#include <stdio.h>

int main(){
    void printN(int);
    int n;

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

    printN(n);
    return 0;
}

void printN(int n){
    if(n == 0){
        return;
    }
    printf("%d ", n);
    printN(n - 1);
}
```

## 6. print using 1 to n using recursion. ↗

Test Data

Enter the value of n: 5



Expected Output

1 2 3 4 5



Source Code

```
#include <stdio.h>

int main(){
    void printN(int);
    int n;

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

    printN(n);
    return 0;
}

void printN(int n){
    if(n == 0){
        return;
    }
    printN(n - 1);
    printf("%d ", n);
}
```



## 7. Recursive Program to print multiplication table of a number ↗

Test Data

Enter the value of n: 5



Expected Output

5 X 1 = 5  
5 X 2 = 10  
5 X 3 = 15  
5 X 4 = 20  
5 X 5 = 25  
5 X 6 = 30  
5 X 7 = 35  
5 X 8 = 40  
5 X 9 = 45  
5 X 10 = 50



Source Code

```
#include <stdio.h>

int main(){
    void printTable(int, int);
    int n;

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

    printTable(n, 1);
    return 0;
}

void printTable(int n, int t){
    if(t > 10){
        return;
    }
    printf("\n%d X %d = %d", n, t, n * t);
    printTable(n, t + 1);
}
```

## 8. ↗

Test Data

Expected Output

Source Code



# Arrays ↗

## 1. WAP. to print sum of all elements. ↗

Test Data

```
Enter length of the Array: 5

Enter Element 1: 5

Enter Element 2: 10

Enter Element 3: 2

Enter Element 4: 3

Enter Element 5: 4
```



### Expected Output

```
Sum = 24
```



### Source Code

```
#include <stdio.h>

int calcSum(int arr[], int length){
    int sum = 0, i;
    for(i = 0; i < length; i++){
        sum += arr[i];
    }
    return sum;
}

int main(){
    int n, i, sum;
    int arr[100];

    printf("\nEnter length of the Array: ");
    scanf("%d", &n);

    for(i = 0; i < n; i++){
        printf("\nEnter Element %d: ", i + 1);
        scanf("%d", &arr[i]);
    }

    sum = calcSum(arr, n);

    printf("\nSum = %d", sum);
    return 0;
}
```



## 2. WAP. to display the sum of all odd and even numbers. ↗

### Test Data

```
Enter any 10 numbers: 10 15 20 13 30 20 55 22 33 29
```



### Expected Output

```
Sum of Even number = 102 and Odd number = 145
```



### Source Code

```
#include <stdio.h>

void evenOddSum(int arr[], int length){
    int evenSum = 0, oddSum = 0;
    for(int i = 0; i < length; i++){
        if(arr[i] % 2 == 0){
            evenSum += arr[i];
        }
        else{
            oddSum += arr[i];
        }
    }
    printf("Sum of Even number = %d and Odd number = %d", evenSum, oddSum);
}
```



```

    }
    else{
        oddSum += arr[i];
    }
}
printf("\nSum of Even number = %d and Odd number = %d", evenSum, oddSum);
}

int main(){
    int arr[100];
    int n;

    printf("\nEnter length of the array: ");
    scanf("%d", &n);

    for(int i = 0; i < n; i++){
        printf("\nEnter Element %d: ", i + 1);
        scanf("%d", &arr[i]);
    }

    evenOddSum(arr, n);
    return 0;
}

```

### 3. Program to find the average of n numbers using arrays ↗

Test Data

```
Enter number of elements: 5
Enter marks: 50 75 85 65 92
```



Expected Output

```
Average marks = 73.40
```



Source Code

```
#include <stdio.h>

void calcAverage(float arr[], int n){
    float sum = 0, average;

    for(int i = 0; i < n; i++){
        printf("\nEnter marks %d: ", i + 1);
        scanf("%d", &arr[i]);
    }

    for(int i = 0; i < n; i++)
        sum += arr[i];

    average = sum / n;

    printf("Average marks = %.2f", average);
}
```

```
int main(){
    int n;
    float arr[100];

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

    calcAverage(arr, n);
}
```

#### 4. WAP. to find out the Largest element in the array. ↗

##### Test Data

```
Enter number of elements: 5

Enter element 1: 55

Enter element 2: 12

Enter element 3: 22

Enter element 4: 78

Enter element 5: 23
```



##### Expected Output

```
Largest element = 78
```



##### Source Code

```
#include <stdio.h>

int findMax(int arr[], int length){
    int i, max = 0;
    for(i = 0; i < length; i++){
        if(max < arr[i]){
            max = arr[i];
        }
    }
    return max;
}

int main(){
    int n, max;
    int arr[100];

    printf("\nEnter number of elements: ");
    scanf("%d", &n);

    for(int i = 0; i < n; i++){
        printf("\nEnter element %d: ", i + 1);
        scanf("%d", &arr[i]);
    }
}
```



```
    max = findMax(arr, n);

    printf("\nLargest element = %d", max);
}
```

## 5. WAP. to find out the Smallest element in the array. ↗

### Test Data

```
Enter number of elements: 5

Enter element 1: 22

Enter element 2: 55

Enter element 3: 59

Enter element 4: 75

Enter element 5: 21
```



### Expected Output

```
Smallest element = 21
```



### Source Code

```
#include <stdio.h>

int findMin(int arr[], int length){
    int i, min = arr[0];
    for(i = 0; i < length; i++){
        if(min > arr[i]){
            min = arr[i];
        }
    }
    return min;
}

int main(){
    int n, min;
    int arr[100];

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

    for(int i = 0; i < n; i++){
        printf("\nEnter element %d: ", i + 1);
        scanf("%d", &arr[i]);
    }

    min = findMin(arr, n);
```



```
    printf("\nSmallest element = %d", min);
}
```

## 6. Write a C program to print all negative elements in an array. ↗

Test Data

```
Enter length of the array: 5
Enter 5 elements: 1 2 -5 2 -1
```



Expected Output

```
-5 -1
```



Source Code

```
#include <stdio.h>

int main(){
    int arr[100];
    int i, length;

    printf("Enter length of the array: ");
    scanf("%d", &length);

    printf("Enter %d Elements\n", length);
    for(i = 0; i < length; i++){
        scanf("%d", &arr[i]);
    }

    for(i = 0; i < length; i++){
        if(arr[i] < 0){
            printf("%d ", arr[i]);
        }
    }

    return 0;
}
```



## 7. Write C program to count total number of negative elements in array. ↗

Test Data

```
Enter length of the array: 5
Enter 5 Element: 1 2 -3 5 -4
```



Expected Output

```
Total negative number = 2
```



## Source Code

```
#include <stdio.h>

int main(){
    int arr[100];
    int i, length, count = 0;

    printf("Enter length of the array: ");
    scanf("%d", &length);

    printf("Enter %d Element\n", length);
    for(i = 0; i < length; i++){
        scanf("%d", &arr[i]);
    }

    for(i = 0; i < length; i++){
        if(arr[i] < 0){
            count++;
        }
    }

    printf("Total negative number = %d", count);
    return 0;
}
```



## 8. C program to insert an element at end of an Array ↗

### Test Data

```
Enter length of the array: 5
Enter 5 element: 1 2 3 4 5
Give a number to insert at end: 6
```



### Expected Output

```
1 2 3 4 5 6
```



## Source Code

```
#include <stdio.h>

int main(){
    int arr[100];
    int i, length, num;

    printf("Enter length of the array: ");
    scanf("%d", &length);

    printf("Enter %d Element\n", length);
    for(i = 0; i < length; i++){
        scanf("%d", &arr[i]);
    }

    printf("Give a number to insert at end: ");
```



```
scanf("%d", &num);
length++;
arr[length-1] = num;

for(i = 0; i < length;i++){
    printf("%d ", arr[i]);
}
return 0;
}
```

## 9. Write C program to insert an element at beginning of an array ↗

### Test Data

```
Enter length of the array: 5
Enter 5 element: 1 2 3 4 5
Give a number to insert at beginning: 0
```



### Expected Output

```
0 1 2 3 4 5
```



### Source Code

```
#include <stdio.h>

int main(){
    int arr[100];
    int i, length, num;

    printf("Enter length of the array: ");
    scanf("%d", &length);

    printf("Enter %d Element\n", length);
    for(i = 0; i < length; i++){
        scanf("%d", &arr[i]);
    }
    printf("Give a number to insert at beginning: ");
    scanf("%d", &num);
    length++;

    for(i = length - 1; i >= 0; i--){
        arr[i + 1] = arr[i];
    }
    arr[0] = num;

    for(i = 0; i < length;i++){
        printf("%d ", arr[i]);
    }
    return 0;
}
```

## 10. Write C program to insert an element in array. ↗

Test Data

```
Enter length of the array: 3
Enter 3 elements: 1 2 4
Enter the position: 3
Enter the value: 3
```



Expected Output

```
1 2 3 4
```



Source Code

```
#include <stdio.h>

int main(){
    int arr[100];
    int length, i, position, value;

    printf("Enter length of the array: ");
    scanf("%d", &length);

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

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

    if(position < 1 || position > length + 1){
        printf("Enter Valid position..!");
        return 0;
    }

    for(i = length; i >= position; i--){
        arr[i] = arr[i - 1];
    }
    length++;
    arr[position - 1] = value;

    for(i = 0; i < length; i++){
        printf("%d ", arr[i]);
    }
    return 0;
}
```



## 11. Write C program to print all unique element in an array. ↗

Test Data

Enter length of the array: 4  
Enter 4 elements: 3 2 2 5



#### Expected Output

The unique elements found in the array are: 3 5



#### Source Code

```
#include <stdio.h>

int main(){
    int arr[100];
    int length, i, j, f;

    printf("Enter length of the array: ");
    scanf("%d", &length);

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

    printf("\nThe unique elements found in the array are: ");
    for(i = 0; i < length; i++){
        f = 0;
        for(j = 0; j < length; j++){
            if(arr[i] == arr[j] && i != j){
                f++;
            }
        }
        if(f == 0){
            printf("%d ", arr[i]);
        }
    }

    return 0;
}
```



## 12. Write C program to sort an array in ascending order. ↗

#### Test Data

Enter length of the array: 5  
Enter 5 elements: 2 1 5 4 3



#### Expected Output

1 2 3 4 5



#### Source Code

```

#include <stdio.h>

int main(){
    int arr[100];
    int length, i, j, temp;

    printf("Enter length of the array: ");
    scanf("%d", &length);

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

    // Selection Sort
    // for(i = 0; i < length - 1; i++){
    //     for(j = i + 1; j < length; j++){
    //         if(arr[i] > arr[j]){
    //             temp = arr[i];
    //             arr[i] = arr[j];
    //             arr[j] = temp;
    //         }
    //     }
    // }

    // Bubble Sort
    // for(i = 1; i < length; i++){
    //     for(j = 0; j < length - i; j++){
    //         if(arr[j] > arr[j + 1]){
    //             temp = arr[j];
    //             arr[j] = arr[j + 1];
    //             arr[j + 1] = temp;
    //         }
    //     }
    // }

    // Insertion Sort
    for(i = 1; i < length; i++){
        int current = arr[i];
        j = i - 1;
        while(arr[j] > current && j >= 0){
            arr[j + 1] = arr[j];
            j--;
        }
        arr[j + 1] = current;
    }

    for(i = 0; i < length; i++){
        printf("%d ", arr[i]);
    }

    return 0;
}

```



### 13. Write C program to copy all elements of one array to another. ↗

Test Data

```
Enter length of the array: 5
Enter 5 elements: 1 2 3 4 5
```



### Expected Output

```
1st Array: 1 2 3 4 5
2nd Array: 1 2 3 4 5
```



### Source Code

```
#include <stdio.h>

void copyArray(int arr[], int arr2[], int length){
    int i;
    for(i = 0; i < length; i++){
        arr2[i] = arr[i];
    }
}

void printArray(int arr[], int length){
    for(int i = 0; i < length; i++){
        printf("%d ", arr[i]);
    }
}

int main(){
    int arr[100];
    int arr2[100];
    int length, i;

    printf("Enter length of the array: ");
    scanf("%d", &length);

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

    copyArray(arr, arr2, length);

    printf("\n1st Array: ");
    printArray(arr, length);
    printf("\n2nd Array: ");
    printArray(arr2, length);
    return 0;
}
```



## 14. Write C program to delete an element from an array ↗

### Test Data

```
Enter length of the array: 5
```



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

Expected Output

```
1 2 3 4
```



Source Code

```
#include <stdio.h>

int main(){
    int arr[100];
    int length, i, position;

    printf("Enter length of the array: ");
    scanf("%d", &length);

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

    printf("Enter position of element you want to delete: ");
    scanf("%d", &position);

    for(i = position - 1; i < length - 1;i++){
        arr[i] = arr[i + 1];
    }
    length--;

    for(i = 0; i < length; i++){
        printf("%d ", arr[i]);
    }
    return 0;
}
```



## 15. Write C program to delete all duplicate elements from an array ↗

Test Data

```
Enter the length of the array: 10
Enter 10 elements: 57 12 89 32 62 12 89 35 67 75
```



Expected Output

```
57 12 89 32 62 35 67 75
```



Source Code

```
#include <stdio.h>

int main(){
```



```

int arr[100];
int length, i, j, k, position;

printf("Enter length of the array: ");
scanf("%d", &length);

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

for(i = 0; i < length; i++){
    for(j = 0; j < length; j++){
        if(arr[i] == arr[j] && i != j){
            int current = j;
            for(k = current; k < length; k++){
                arr[k] = arr[k + 1];
            }
            length--;
        }
    }
}

for(i = 0; i < length; i++){
    printf("%d ", arr[i]);
}
return 0;
}

```

## 16. Write C program to count number of each element in an array. ↗

Test Data

Enter 10 elements: 57 12 89 32 62 12 89 35 67 75



Expected Output

57 occurs 1 times  
 12 occurs 2 times  
 89 occurs 2 times  
 32 occurs 1 times  
 62 occurs 1 times  
 35 occurs 1 times  
 67 occurs 1 times  
 75 occurs 1 times



Source Code

```

#include <stdio.h>

int main(){
    int arr[100];
    int length, i, j, k, position;

    printf("Enter length of the array: ");

```

```

scanf("%d", &length);

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

for(i = 0; i < length; i++){
    int count = 1;
    for(j = 0; j < length; j++){
        if(arr[i] == arr[j] && i != j){
            count++;
            int current = j;
            for(k = current; k < length; k++){
                arr[k] = arr[k + 1];
            }
            length--;
        }
    }
    printf("\n%d occurs %d times", arr[i], count);
}

return 0;
}

```

## 17. Write C program count total duplicate elements in an array. ↗

### Test Data

```

Enter length of the array: 10
Enter 10 elements: 1 1 1 2 2 3 4 3 1 5

```



### Expected Output

```

Number of duplicate elements count = 3

```



### Source Code

```

#include <stdio.h>

int main(){
    int arr[100];
    int length, i, count = 0, position, dup;

    printf("Enter length of the array: ");
    scanf("%d", &length);

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

    for(i = 0; i < length; i++){
        for(int j = 0; j < length; j++){
            if(arr[j] == arr[i] && i != j){

```

```

        if(dup != arr[j]){
            count++;
            dup = arr[j];
        }
        position = j;
        for(int k = position; k < length; k++){
            arr[k] = arr[k + 1];
        }
        length--;
    }
}

printf("Number of duplicate elements count = %d", count);
}

```

## 18. Write C program to merge two sorted array ↗

Array

```
Array 1 = {9, 7, 3, 5, 1};
Array 2 = {4, 6, 8, 10};
```

Expected Output

```
1 3 4 5 6 7 8 9 10
```

Source Code

```
#include <stdio.h>

void sort(int arr[], int length){
    int i, j;
    for(i = 0; i < length; i++){
        for(j = 0; j < length - 1;j++){
            if(arr[j] > arr[j+1]){
                int temp = arr[j];
                arr[j] = arr[j+1];
                arr[j+1] = temp;
            }
        }
    }
}

int main(){
    int length1 = 5, length2 = 4;
    int arr1[5] = {9, 7, 3, 5, 1};
    int arr2[4] = {4, 6, 8, 10};
    int arr3[length1+length2];
    int i, j = 0, k = 0;

    sort(arr1, length1);
    sort(arr2, length2);

    for(i = 0; k < length2 || j < length1; i++){
        if(j < length1 && arr1[j] < arr2[k]){
            arr3[i] = arr1[j];
            j++;
        } else {
            arr3[i] = arr2[k];
            k++;
        }
    }
}
```

```

        if(arr1[j] > arr2[k]){
            arr3[i] = arr2[k];
            k++;
        } else{
            arr3[i] = arr1[j];
            j++;
        }
    }

    for(i = 0; i < length1 + length2; i++){
        printf("%d ", arr3[i]);
    }

    return 0;
}

```

## 19. Write C program to put even and odd elements of array in two separate array



Array

{55, 13, 22, 45, 67, 68, 88, 98, 19, 12}



Expected Output

Even Array: 22 68 88 98 12  
 Odd Array: 55 13 45 67 19



Source Code

```

#include <stdio.h>

void printArray(int arr[], int length){
    for(int i = 0; i < length; i++){
        printf("%d ", arr[i]);
    }
}

int isEven(int num){
    if(num % 2 == 0){
        return 1;
    }
    return 0;
}

int main(){
    int arr[10] = {55, 13, 22, 45, 67, 68, 88, 98, 19, 12};
    int length = 10, i, j = 0, k = 0;
    int evenArray[100];
    int oddArray[100];

    for(i = 0; i < length; i++){
        if(isEven(arr[i])){
            evenArray[j] = arr[i];
            j++;
        }
    }
}

```

```

        } else{
            oddArray[k] = arr[i];
            k++;
        }
    }

    printf("\nEven Array: ");
    printArray(evenArray, j);

    printf("\nOdd Array: ");
    printArray(oddArray, k);

    return 0;
}

```

## 20. Write C program to find reverse of an array. ↗

Array

{55, 13, 22, 45, 67, 68, 88, 98, 19, 12}



Expected Output

12 19 98 88 68 67 45 22 13 55



Source Code

```

#include <stdio.h>

void swap(int arr[], int i, int j){
    int temp = arr[i];
    arr[i] = arr[j];
    arr[j] = temp;
}

int main(){
    int arr[5] = {1, 2, 3, 4, 5};
    int length = 5, i;

    for(i = 0; i < length / 2; i++){
        swap(arr, i, length - i - 1);
    }

    for(i = 0; i < length; i++){
        printf("%d ", arr[i]);
    }
    return 0;
}

```

## 21. Write C program to right rotate an array. ↗

Array

```
{1, 2, 3, 4, 5}
```



### Test Data

```
Enter the value of k: 2
```



### Expected Output

```
4 5 1 2 3
```



### Source Code

```
#include <stdio.h>

void reverse(int arr[], int i, int j){
    while(i < j){
        int temp = arr[i];
        arr[i] = arr[j];
        arr[j] = temp;
        i++;
        j--;
    }
}

void printArray(int arr[], int n){
    for(int i = 0; i < n; i++){
        printf("%d ", arr[i]);
    }
}

int main(){
    int arr[5] = {1, 2, 3, 4, 5};
    int n = 5, k;

    printf("Enter the value of k: ");
    scanf("%d", &k);
    k = k % n;
    reverse(arr, n - k, n - 1);
    reverse(arr, 0, n - k - 1);
    reverse(arr, 0, n - 1);
    printArray(arr, n);

    return 0;
}
```



## 22. Write C program to left rotate an array. ↗

### Array

```
{1, 2, 3, 4, 5}
```



### Test Data

Enter the value of k: 2



### Expected Output

3 4 5 1 2



### Source Code

```
#include <stdio.h>

void reverse(int arr[], int i, int j){
    while(i < j){
        int temp = arr[i];
        arr[i] = arr[j];
        arr[j] = temp;
        i++;
        j--;
    }
}

void printArray(int arr[], int n){
    for(int i = 0; i < n; i++){
        printf("%d ", arr[i]);
    }
}

int main(){
    int arr[5] = {1, 2, 3, 4, 5};
    int n = 5, k;

    printf("Enter the value of k: ");
    scanf("%d", &k);
    k = k % n;
    reverse(arr, 0, k - 1);
    reverse(arr, k, n - 1);
    reverse(arr, 0, n - 1);
    printArray(arr, n);

    return 0;
}
```



Hi 🙌, I'm Rajiv Kumar ☺

A passionate frontend developer from India ☺

## Connect with me: ↗

 GITHUB

 TWITTER

 LINKEDIN

## Tech Stack: ↗

   