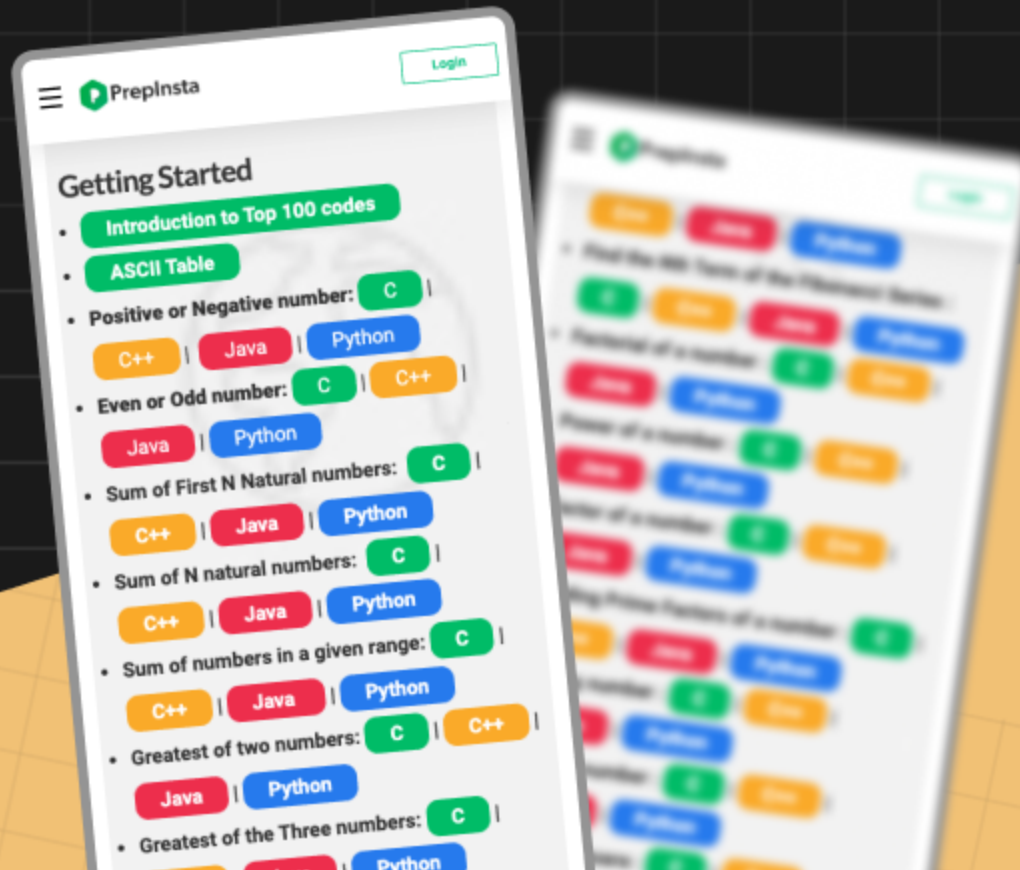


# Top 100 code to Prepare for Placements Solution in all 4 language



## Q1 : Positive or Negative number

Given an integer input, The objective is to write a code to Check if a Number is Positive or Negative

### Code in C

```
#include <stdio.h>
int main()
{
    int num = 23;

    //Conditions to check if the number is negative/positive or zero
    if (num > 0)
        printf("The number is positive");
    else if (num < 0)
        printf("The number is negative");
    else
        printf("Zero");

    return 0;
}
```

### Code in C++

```
#include <iostream>
using namespace std;

int main()
{
    int num = 96;

    //Conditions to check if the number is negative or positive
    if (num > 0)
        cout << "The number is positive";
    else if (num < 0)
        cout << "The number is negative";
    else
        cout << "Zero";

    return 0;
}
```

```
}
```

## Code in Java

```
class Main
{
    public static void main (String[]args)
    {

        int num = 5;

        //Conditions to check if the number is negative or positive
        if (num > 0)
            System.out.println ("The number is positive");
        else if (num < 0)
            System.out.println ("The number is negative");
        else
            System.out.println ("Zero");
    }
}
```

## Code in Python

```
num = 15
if num > 0:
    print('Positive')
elif num < 0:
    print('Negative')
else:
    print('Zero')
```

## Q2: Even or Odd number

Write a program to check whether a number is Even or Odd program

## Code in C

```
#include <stdio.h>

int main ()
{
    int number;
```

```

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

//Checking if the number is divisible by 2
if (number % 2 == 0)
printf ("Even");
else
printf ("Odd");

return 0;
}

```

## Code in C++

```

#include <iostream>
using namespace std;
int main ()
{
    int number;
    cout << "Enter a number:"; cin >> number;

    //checking whether the number is even or odd
    if (number % 2 == 0)
    cout << number << " : Even";
    else
    cout << number << " : Odd";

    return 0;
}

```

## Code in Java

```

public class Main
{
    public static void main(String[] args) {
        int number = 29;

        //checking whether the number is even or odd
        if (number % 2 == 0)
        System.out.println(number + " is Even");
        else
        System.out.println(number + " is odd");
    }
}

```

## Code in Python

```
num = int(input("Enter a Number:"))
if num % 2 == 0:
    print("Given number is Even")
else:
    print("Given number is Odd")
```

## Q3: Sum of First N Natural numbers

Write a program to find the sum of first N natural numbers.

## Code in C

```
#include <stdio.h>

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

    int sum = 0;

    for(int i=1;i<=n;i++)
        // is same as writing sum = sum + i
        sum += i;

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

    return 0;
}
```

## Code in C++

```
#include<bits/stdc++.h>
using namespace std;

int main()
{
    int n;
```

```
    cout << "Enter a number : ";  
    cin >> n;  
  
    int sum=0;  
  
    for(int i=1;i<=n;i++)  
        sum+=i;  
  
    cout << sum;  
  
    return 0;  
}
```

## Code in Java

```
public class Main  
{  
    public static void main (String[]args)  
    {  
  
        int n = 10;  
        int sum = 0;  
  
        for (int i = 1; i <= n; i++)  
            sum += i;  
        System.out.println (sum);  
    }  
}
```

## Code in Python

```
num = 5  
sum = 0  
for i in range(num+1):  
    sum+=i  
print(sum)
```

## Q4: Sum of numbers in a given range

Given two integer inputs num1 and num2, the objective is to write a code to Find the Sum of Numbers in a Given Range

### Code in C

```
#include <stdio.h>

int main()
{
    int a = 5;
    int b = 10;

    int sum = 0;

    for (int i = a; i <= b; i++)
        sum = sum + i;

    printf("%d",sum);

    return 0;
}
```

### Code in C++

```
#include<bits/stdc++.h>
using namespace std;
int main()
{
    int num1 = 5;
    int num2 = 10;
    int sum = 0;
    for (int i = num1; i <= num2; i++)
        sum = sum + i;
    cout << sum;
    return 0;
}
```

### Code in Java

```
public class Main
{
    public static void main (String[]args)
    {
```

```

int a = 5;
int b = 10;

int sum = 0;

for (int i = a; i <= b; i++)
    sum = sum + i;
System.out.println ("The sum is " + sum);
}
}

```

## Code in Python

```

num1, num2 = 3, 6
sum = 0
for i in range(num1,num2+1):
    sum+=i
print(sum)

```

## Q5: Greatest of two numbers

Write a program to find the greatest of two numbers

## Code in C

```

#include<stdio.h>

int main ()
{
    int num1, num2;
    num1=12,num2=13;

    if (num1 == num2)
        printf("both are equal");
    else if (num1 > num2)
        printf("%d is greater", num1);
    else
        printf("%d is greater", num2);

    return 0;
}

```



## Code in C++

```
#include <iostream>
using namespace std;

int main ()
{
    int num1, num2;
    num1=75,num2=85;

    if (num1 == num2)
        cout << "both are equal"; else if (num1 > num2)
        cout << num1 << " is greater than " << num2;
    else
        cout << num2 << " is greater than " << num1;

    return 0;
}
```

## Code in Java

```
public class Main
{
    public static void main (String[]args)
    {

        int num1 = 50, num2 = 20;
        if (num1 == num2)
            System.out.println ("both are equal");
        else if (num1 > num2)
            System.out.println (num1 + " is greater");

        else
            System.out.println (num2 + " is greater");

    }
}
```

## Code in Python

```
num1, num2 = 20 , 30
if num1>num2:
    print(num1)
else:
    print(num2)
```

## Q6: Leap year or not

Write a program to check whether a year is leap year or not

### Code in C

```
#include <stdio.h>
int main ()
{
    int year;
    year=2000;

    if(year % 400 == 0)
        printf("%d is a Leap Year",year);

    else if(year % 4 == 0 && year % 100 != 0)
        printf("%d is a Leap Year",year);

    else
        printf("%d is not a Leap Year",year);

    return 0;
}
```

### Code in C++

```
#include <bits/stdc++.h>
using namespace std;

int main()
{
    int year;

    year=2000;

    if(year % 400 == 0)
        cout << year << " is a Leap Year";

    else if(year % 4 == 0 && year % 100 != 0)
        cout << year << " is a Leap Year";

    else
        cout << year << " is not a Leap Year";
}
```

```
        return 0;
    }
```

## Code in Java

```
public class Main{
    public static void main (String[]args)
    {

        int year = 2020;

        if (year % 400 == 0)
            System.out.println (year + " is a Leap Year");

        else if (year % 4 == 0 && year % 100 != 0)
            System.out.println (year + " is a Leap Year");

        else
            System.out.println (year + " is not a Leap Year");

    }
}
```

## Code in Python

```
year = 2000

if (year%400 == 0) or (year%4==0 and year%100!=0):
    print("Leap Year")
else:
    print("Not a Leap Year")
```

## Q7: Prime number within a given range

Write a code to find the prime numbers between two numbers.

## Code in C

```
#include <stdio.h>

int checkPrime(int num)
```

```

{
    // 0, 1 and negative numbers are not prime
    if(num < 2){
        return 0;
    }
    else{
        // no need to run loop till num-1 as for any number x the numbers in
        // the range(num/2 + 1, num) won't be divisible anyways.
        // Example 36 wont be divisible by anything b/w 19-35
        int x = num/2;
        for(int i = 2; i < x; i++)
        {
            if(num % i == 0)
            {
                return 0;
            }
        }
        // the number would be prime if we reach here
        return 1;
    }
}

```

```

int main()
{
    int a=10, b=20;

    for(int i=a; i <= b; i++){
        if(checkPrime(i))
            printf("%d ",i);
    }

    return 0;
}

```

## Code in C++

```

#include<bits/stdc++.h>
using namespace std;

```

```

bool isPrime(int n){
    int count = 0;

    // 0, 1 negative numbers are not prime
    if(n < 2)

```

```

        return false;

        // checking the number of divisors b/w 1 and the number n-1
        for(int i = 2; i < n; i++)
        {
            if(n % i == 0)
                return false;
        }

        // if reached here then must be true
        return true;
    }

int main()
{
    int lower, upper;

    lower=1,upper=100;

    for(int i = lower; i <= upper; i++)
        if(isPrime(i))
            cout << i << " ";

}

```

## Code in Java

```

public class Main
{
    public static void main (String[]args)
    {

        int lower = 1, upper = 20;

        for (int i = lower; i <= upper; i++)
            if (isPrime (i))
                System.out.println (i);
    }

    static boolean isPrime (int n)
    {
        int count = 0;

        // 0, 1 negative numbers are not prime

```

```

    if (n < 2)
        return false;

    // checking the number of divisors b/w 1 and the number n-1
    for (int i = 2; i < n; i++)
    {
        if (n % i == 0)
            return false;
    }

    // if reached here then must be true
    return true;
}
}

```

## Code in Python

```

low, high = 2, 10
primes = []

for i in range(low, high + 1):
    flag = 0

    if i < 2:
        continue
    if i == 2:
        primes.append(2)
        continue

    for x in range(2, i):
        if i % x == 0:
            flag = 1
            break

    if flag == 0:
        primes.append(i)

print(primes)

```

## Q8: Reverse of a number

Write a program to reverse a number.

## Code in C

```
#include<stdio.h>

//main program
int main ()
{
    //variables initialization
    int num, reverse = 0, rem;
    num=1234;
    printf("The number is: %d\n",num);

    //loop to find reverse number
    while(num != 0)
    {
        rem = num % 10;
        reverse = reverse * 10 + rem;
        num /= 10;
    };

    //output
    printf("Reverse: %d\n",reverse);

    return 0;
}
```

## Code in C++

```
#include <iostream>
using namespace std;

//main program
int main ()
{
    //variables initialization
    int num, reverse = 0, rem;
    num=1234;
    cout <<"\nThe number is"<<num;

    //loop to find reverse number
    while(num != 0)
    {
        rem = num % 10;
```

```

        reverse = reverse * 10 + rem;
        num /= 10;
    };

    //output
    cout <<"\nReversed Number: "<<reverse;

    return 0;
}

```

## Code in Java

```

public class Main
{
    public static void main (String[]args)
    {

        //variables initialization
        int num = 1234, reverse = 0, rem;

        //loop to find reverse number
        while (num != 0)
        {
            rem = num % 10;
            reverse = reverse * 10 + rem;
            num /= 10;
        };

        //output
        System.out.println ("Reversed Number: " + reverse);
    }
}

```

## Code in Python

```

num = 1234
temp = num
reverse = 0
while num > 0:
    remainder = num % 10
    reverse = (reverse * 10) + remainder
    num = num // 10

```



```
print(reverse)
```

## Q9: Armstrong number

Write a program to check whether a number is armstrong or not.

### Code in C

```
#include <stdio.h>
#include <math.h>
int order(int x)
{
    int len = 0;
    while (x)
    {
        len++;
        x = x/10;
    }
    return len;
}
int getArmstrongSum(int num, int order){

    if(num == 0)
        return 0;

    int digit = num % 10;

    return pow(digit, order) + getArmstrongSum(num/10, order);
}

// Driver Code
int main ()
{
    int num, len;
    num=1634;
    printf("The number is:%d\n",num);

    // function to get order(length)
    len = order(num);

    // check if Armstrong
    if (num == getArmstrongSum(num, len))
```

```

        printf("%d is Armstrong", num);
    else
        printf("%d is Not Armstrong", num);

}

```

## Code in C++

```

#include<bits/stdc++.h>
#include<math.h>
using namespace std;

```

```

int order(int x)

```

```

{
    int len = 0;
    while (x)
    {
        len++;
        x = x/10;
    }
    return len;
}

```

```

bool armstrong(int num, int len){

```

```

    int sum = 0, temp, digit;
    temp = num;

```

```

    // loop to extract digit, find power & add to sum
    while(temp != 0)
    {
        // extract digit
        digit = temp % 10;

```

```

        // add power to sum
        sum = sum + pow(digit,len);
        temp /= 10;
    };

```

```

    return num == sum;

```

```

}

```

```

// Driver Code

```

```

int main ()

```

```

{

```

```

//variables initialization
int num = 407, len;

// function to get order(length)
len = order(num);

// check if Armstrong
if (armstrong(num, len))
    cout << num << " is armstrong";
else
    cout << num << " is not armstrong";

return 0;
}

```

## Code in Java

```

public class Main
{
    public static void main (String[] args)
    {
        int num = 407, len;

        // function to get order(length)
        len = order (num);

        // check if Armstrong
        if (armstrong (num, len))
            System.out.println(num + " is armstrong");
        else
            System.out.println(num + " is armstrong");
    }

    static int order (int x)
    {
        int len = 0;
        while (x != 0 )
        {
            len++;
            x = x / 10;
        }
        return len;
    }
}

```

```

}

static boolean armstrong (int num, int len)
{
    int sum = 0, temp, digit;
    temp = num;

    // loop to extract digit, find power & add to sum
    while (temp != 0)
    {
        // extract digit
        digit = temp % 10;

        // add power to sum
        sum = sum + (int)Math.pow(digit, len);
        temp /= 10;
    };

    return num == sum;
}
}

```

## Code in Python

```

number = 371
num = number
digit, sum = 0, 0
length = len(str(num))
for i in range(length):
    digit = int(num%10)
    num = num/10
    sum += pow(digit,length)
if sum==number:
    print("Armstrong")
else:
    print("Not Armstrong")

```

## Q10: Highest Common Factor(HCF)

Write a program to find HCF of two numbers

## Code in C

```
#include<stdio.h>

int main()
{
    int num1 = 36, num2 = 60, hcf = 1;

    for(int i = 1; i <= num1 || i <= num2; i++) {
        if(num1 % i == 0 && num2 % i == 0)
            hcf = i;
    }

    printf("The HCF: %d", hcf);

    return 0;
}
```

## Code in C++

```
#include<iostream>
using namespace std;

int main()
{
    int num1 = 36, num2 = 60, hcf = 1;

    for(int i = 1; i <= num1 || i <= num2; i++)
    {
        if(num1 % i == 0 && num2 % i == 0)
            hcf = i;
    }

    cout<<"HCF of "<<num1<<" and "<<num2<<" is "<<hcf;

    return 0;
}
```

## Code in Java

```
class Main
{
    public static void main (String[]args)
    {
        int num1 = 36, num2 = 60, hcf=0;
```

```

for (int i = 1; i <= num1 || i <= num2; i++)
{
    if (num1 % i == 0 && num2 % i == 0)
        hcf = i;
}

System.out.println("The HCF: "+ hcf);
}
}

```

## Code in Python

```

num1 = 36
num2 = 60
hcf = 1

for i in range(1, min(num1, num2)):
    if num1 % i == 0 and num2 % i == 0:
        hcf = i
print("Hcf of", num1, "and", num2, "is", hcf)

```

## Q11: Lowest Common Multiple (LCM)

Write a program to find LCM of two numbers.

### Code in C

```

#include<stdio.h>

int main()
{
    int num1 = 36, num2 = 60, lcm;

    // finding the larger number here
    int max = (num1 > num2)? num1 : num2;

    // LCM will atleast be >= max(num1, num2)
    // Largest possibility of LCM will be num1*num2
    for(int i = max ; i <= num1*num2 ; i++)
    {
        if(i % num1 == 0 && i % num2 == 0){
            lcm = i;
            break;
        }
    }
}

```

```

    }
}

printf("The LCM: %d", lcm);

return 0;
}

```

## Code in C++

```

#include<bits/stdc++.h>
using namespace std;

int main()
{
    int num1 = 12, num2 = 14, lcm;

    // finding the larger number here
    int max = (num1 > num2)? num1 : num2;

    // LCM will atleast be >= max(num1, num2)
    // Largest possibility of LCM will be num1*num2
    for(int i = max ; i <= num1*num2 ; i++)
    {
        if(i % num1 == 0 && i % num2 == 0){
            lcm = i;
            break;
        }
    }

    cout<<"LCM of "<<num1<<" and "<<num2<<" is "<<lcm;

    return 0;
}

```

## Code in Java

```

public class Main
{
    public static void main (String[]args)
    {
        int num1 = 36, num2 = 60, lcm = 0;

        // finding the larger number here
        int max = (num1 > num2) ? num1 : num2;
    }
}

```

```

// LCM will atleast be >= max(num1, num2)
// Largest possibility of LCM will be num1*num2
for (int i = max; i <= num1 * num2; i++)
{
    if (i % num1 == 0 && i % num2 == 0)
    {
        lcm = i;
        break;
    }
}
System.out.println ("The LCM: " + lcm);
}
}

```

## Code in Python

```

num1 = 12
num2 = 14
for i in range(max(num1, num2), 1 + (num1 * num2)):
    if i % num1 == 0 && i % num2 == 0:
        lcm = i
        break
print("LCM of", num1, "and", num2, "is", lcm)

```

## Q12: Factorial of a number

Write a program to find the factorial of a number.

## Code in C

```

#include<stdio.h>
int main ()
{
    int num = 5, fact = 1;

    // Can't calculate factorial of a negative number
    if(num < 0)
        printf("Error");
    else
    {
        for(int i = 1; i <= num; i++)
            fact = fact * i;
    }
}

```



```
        printf("Fact %d: %d",num, fact);  
    }
```

## Code in C++

```
#include<iostream>  
using namespace std;  
int main ()  
{  
    int num = 6, fact = 1;  
  
    // Factorial of negative number doesn't exist  
    // Read more here -  
https://www.quora.com/Is-the-factorial-of-a-negative-number-possible  
    if(num < 0)  
        cout << "Not Possible";  
    else  
    {  
        for(int i = 1; i <= num; i++)  
            fact = fact * i;  
    }  
  
    cout << "Fact " << num << ": " << fact;  
}
```

## Code in Java

```
class Main  
{  
    // Method to find factorial of the given number  
    static int factorial(int n)  
    {  
        int res = 1, i;  
        for (i = 2; i <= n; i++)  
            res *= i;  
        return res;  
    }  
  
    // Driver method  
    public static void main(String[] args)  
    {  
        int num = 6;  
        System.out.println("Factorial of " + num + " is " + factorial(num));  
    }  
}
```

```
}
```

## Code in Python

```
num = 6
fact = 1

# Factorial of negative number doesn't exist

if num < 0:
    print("Not Possible")
else:
    for i in range(1, num+1):
        fact = fact * i

print("Factorial of", num, "is", fact)
```

## Q13: Binary to Decimal conversion

Write a program to convert binary numbers to decimal numbers.

### Code in C

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

int convert(long long num)
{
    int i = 0, decimal= 0;

    while (num!=0)
    {
        int digit = num % 10;
        decimal += digit * pow(2,i);

        num /= 10;
        i++;
    }
    return decimal;
}

// main program
int main()
{
```

```

long long binary;

printf("Enter binary number: ");
scanf("%lld", &binary);

printf("%lld", convert(binary));

return 0;
}

```

## Code in C++

```

#include<bits/stdc++.h>
using namespace std;

int getDecimal(long long num)
{
    int i = 0, decimal= 0;

    while (num!=0)
    {
        int digit = num % 10;
        decimal += digit * pow(2,i);

        num /= 10;
        i++;
    }
    return decimal;
}

// main program
int main()
{
    long long binary = 11011;

    cout << getDecimal(binary);

    return 0;
}

```

## Code in Java

```

import java.util.Scanner;
public class Main
{

```

```

public static void main(String args[])
{
    Scanner sc = new Scanner(System.in);
    System.out.print("Enter a binary number : ");
    int binary = sc.nextInt();
    int decimal = 0;
    int n = 0;
    while(binary > 0)
    {
        int temp = binary%10;
        decimal += temp*Math.pow(2, n);
        binary = binary/10;
        n++;
    }
    System.out.println("Decimal number : "+decimal);
    sc.close();
}
}

```

## Code in Python

```

num = 10
binary_val = num
decimal_val = 0
base = 1

while num > 0:
    rem = num % 10
    decimal_val = decimal_val + rem * base
    num = num // 10
    base = base * 2

print("Binary Number is {}\nDecimal Number is {}".format(binary_val, decimal_val))

```

## Q14: Octal to Decimal conversion

Write a program to convert octal numbers to decimal numbers.

### Code in C

```

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

int convert(long long num)

```

```

{
    int i = 0, decimal = 0;

    int base = 8;

    while (num!=0)
    {
        int digit = num % 10;
        decimal += digit * pow(base, i);

        num /= 10;
        i++;
    }
    return decimal;
}

//main program
int main()
{
    long long octal;

    printf("Enter Octal Number: ");
    scanf("%lld", &octal);

    printf("Decimal: %lld", convert(octal));

    return 0;
}

```

## Code in C++

```

#include<iostream>
#include<math.h>
using namespace std;

int getOctal(long long num)
{
    int i = 0, decimal = 0;

    int base = 8;

    while (num!=0)
    {
        int digit = num % 10;
        decimal += digit * pow(base, i);
    }
}

```

```

        num /= 10;
        i++;
    }
    return decimal;
}

// main program
int main()
{
    long long octal = 462;

    cout << getOctal(octal);

    return 0;
}

```

## Code in Java

```

import java.util.Scanner;
public class Main
{
    public static void main(String args[])
    {
        Scanner sc = new Scanner(System.in);
        //input from user
        System.out.print("Enter a octal number : ");
        int octal = sc.nextInt();
        int decimal = 0;
        int n = 0;
        while(octal > 0)
        {
            int temp = octal % 10;
            decimal += temp * Math.pow(8, n);
            octal = octal/10;
            n++;
        }
        System.out.println("Decimal number : "+decimal);
        sc.close();
    }
}

```

## Code in Python

```

def OctalToDecimal(num):

```

```

decimal_value = 0
base = 1

while num:
    last_digit = num % 10
    num = int(num / 10)
    decimal_value += last_digit * base
    base = base * 8
return decimal_value

```

```

octal = 512
print("The decimal value of",octal, " is",OctalToDecimal(octal))

```

## Q15: Factor of a number

Write a program to find factors of a number

### Code in C

```

#include <stdio.h>
//main Program
int main()
{
    int n = 100;

    printf("Factors of %d are : \n", n);

    // finding and printing factors b/w 1 to num
    for(int i = 1; i <= n; i++)
    {
        // if n is divisible by i, then i is a factor of n
        if(n % i == 0)
            printf("%d, ", i);
    }
}

```

### Code in C++

```

#include <bits/stdc++.h>
using namespace std;

//main Program
int main()

```

```

{
    int num;

    num=100;

    cout << "Factors of " << num << " are: " << endl;

    // finding and printing factors b/w 1 to num
    for(int i = 1; i <= num; i++)
    {
        if(num % i == 0)
            cout << i << " ";
    }
}

```

## Code in Java

```

public class Main
{
    public static void main(String[] args) {

        int num = 10;

        System.out.println( "Factors of " + num + " are " );

        // finding and printing factors b/w 1 to num
        for(int i = 1; i <= num; i++)
        {
            if(num % i == 0)
                System.out.println(i + " ");
        }

    }
}

```

## Code in Python

```

# method to print the divisors
def printDivisors(n) :
    i = 1
    while i <= n :
        if (n % i==0) :
            print (i,end=" ")
        i = i + 1

```



```
# Driver method
print ("The divisors of 100 are: ")
printDivisors(100)
```

## Q16: Hexadecimal to Decimal conversion

Write a program to convert hexadecimal numbers to decimal numbers.

### Code in C

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

int convert(char hex[])
{
    int len = strlen(hex);
    int decimal = 0;

    for (int i = len - 1; i >= 0; i--)
    {
        if (hex[i] >= '0' && hex[i] <= '9')
        {
            int digit = hex[i] - '0';
            decimal += digit * pow(16, len - 1 - i);
        }
        else if (hex[i] >= 'A' && hex[i] <= 'F')
        {
            int digit = hex[i] - 'A' + 10;
            decimal += digit * pow(16, len - 1 - i);
        }
    }
    return decimal;
}

int main()
{
    char hex[20];
    scanf("%s", hex);
    printf("%d", convert(hex));
    return 0;
}
```

## Code in C++

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

using namespace std;

int convert(string num)
{
    int len = num.size();
    int dec = 0, index = 0;

    for(int i = len - 1; i >= 0; i--)
    {
        if (num[i] >= '0' && num[i] <= '9')
        {
            int digit = int(num[i]) - 48;
            dec += digit * pow(16, index);
            index++;
        }

        else if (num[i] >= 'A' && num[i] <= 'F')
        {
            int digit = int(num[i]) - 55;
            dec += digit * pow(16, index);
            index++;
        }
    }
    return dec;
}

int main()
{
    string num;
    cin >> num;

    cout << (convert(num));

    return 0;
}
```

## Code in Java

```
class Main
{
```

```

public static void main (String[]args)
{

    String hex = "C9";
    System.out.println (convert (hex));
}

static int convert(String hex){
    String digits = "0123456789ABCDEF";
    hex = hex.toUpperCase();
    int val = 0;
    for (int i = 0; i < hex.length(); i++)
    {
        char c = hex.charAt(i);
        int d = digits.indexOf(c);
        val = 16*val + d;
    }
    return val;
}
}

```

## Code in Python

```

def convert(hex):
    l = len(hex)
    decimal = 0
    pos = 0
    for i in range(l - 1, -1, -1):
        if '0' <= hex[i] <= '9':
            digit = int(hex[i])
            decimal += digit * pow(16, pos)
            pos += 1

        elif 'A' <= hex[i] <= 'F':
            digit = ord(hex[i]) - 55
            decimal += digit * pow(16, pos)
            pos += 1
    return decimal

hex = "C9"

print("decimal value of", hex, "is", convert(hex))

```

## Q17: Perfect number

Write a program to check whether a number is perfect number. (a perfect number is a positive number that is equal to the sum of all its divisors(excluding itself) excluding itself)

### Code in C

```
#include<stdio.h>

int main ()
{
    int num = 28, sum = 0;
    // iteratively check for all numbers in range [1, 27]
    for(int i = 1; i < num; i++){
        // check if i is a divisor, if yes then add to sum
        if(num % i == 0)
            sum = sum + i;
    }

    if(sum == num)
        printf("%d is a perfect number",num);
    else
        printf("%d is not a perfect number",num);
}
```

### Code in C++

```
#include <iostream>
using namespace std;

int main ()
{
    int n = 28, sum = 0;

    for(int i = 1; i < n; i++){
        if(n % i == 0)
            sum = sum + i;
    }

    if(sum == n)
        cout << n << " is a perfect number";
    else
        cout << n << " is not a perfect number";
}
```

```
}
```

## Code in Java

```
public class Main
{
    public static void main (String[]args)
    {

        int n = 28, sum = 0;

        for (int i = 1; i < n; i++)
        {
            if (n % i == 0)
                sum = sum + i;
        }

        if (sum == n)
            System.out.println (n + " Is a perfect number");
        else
            System.out.println (n + " Is not a perfect number");

    }
}
```

## Code in Python

```
n = 28
sum = 0

for i in range(1, n):
    if n % i == 0:
        sum = sum + i

if sum == n:
    print("The number is a Perfect number")
else:
    print("The number is not a Perfect number")
```

## Q18: Decimal to Binary conversion

Write a program to convert decimal numbers to binary numbers.

## Code in C

```
#include<stdio.h>

void convert(int num)
{
    int binaryArray[32];
    int i = 0;
    while (num > 0) {

        binaryArray[i] = num % 2;
        num = num / 2;
        i++;
    }

    for (int j = i - 1; j >= 0; j--)
        printf("%d",binaryArray[j]);
}

int main()
{
    int n = 11;
    convert(n);
    return 0;
}
```

## Code in C++

```
#include<iostream>
using namespace std;

void convertBinary(int num)
{
    // creating an array to store binary equivalent
    int binaryArray[32];

    // using i to store binary bit at given array position
    int i = 0;
    while (num > 0) {

        // resultant remainder is stored at given array position
        binaryArray[i] = num % 2;
        num = num / 2;
        i++;
    }
}
```

```

        // printing binary array in reverse order
        for (int j = i - 1; j >= 0; j--)
            cout << binaryArray[j];
    }

int main()
{
    int n = 21;
    convertBinary(n);
    return 0;
}

```

## Code in Java

```

public class Main
{
    public static void main(String args[])
    {
        //Decimal Number
        int decimal = 12;
        //integer array for storing binary digits
        int binary[] = new int[20];
        int i = 0;
        //writing logic for the conversion
        while(decimal > 0)
        {
            int r = decimal % 2;
            binary[i++] = r;
            decimal = decimal/2;
        }
        //printing result
        System.out.print("Binary number : ");
        for(int j = i-1 ; j >= 0 ; j--)
            System.out.print(binary[j]+"");
    }
}

```

## Code in Python

```

def convertBinary(num):
    binaryArray = []
    while num>0:
        binaryArray.append(num%2)
        num = num//2

```

```
for j in binaryArray:
    print(j, end="")
```

```
decimal_num = 21
convertBinary(decimal_num)
```

## Q19: Perfect Square

Write a code to check whether a number is perfect square or not

### Code in C

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

int isPerfectSquare(long double x)
{
    if (x >= 0) {

        long long sr = sqrt(x);
        return (sr * sr == x);
    }

    return 0;
}

int main()
{
    long long x = 84;
    if (isPerfectSquare(x)==1)
        printf("True");
    else
        printf("False");
    return 0;
}
```

### Code in C++

```
#include <bits/stdc++.h>
using namespace std;

bool isPerfectSquare(long double x)
{
```



```

    if (x >= 0) {

        long long sr = sqrt(x);
        return (sr * sr == x);
    }

    return false;
}

int main()
{
    long long x = 84;
    if (isPerfectSquare(x))
        cout << "True";
    else
        cout << "False";
    return 0;
}

```

## Code in Java

```

public class Main {

    static boolean isPerfectSquare(int x)
    {
        if (x >= 0) {

            int sr = (int)Math.sqrt(x);

            return ((sr * sr) == x);
        }
        return false;
    }

    public static void main(String[] args)
    {
        int x = 84;

        if (isPerfectSquare(x))
            System.out.print("True");
        else
            System.out.print("False");
    }
}

```

## Code in Python

```
from math import sqrt
```

```
def isPerfectSquare(x):  
    if x >= 0:  
        sr = int(sqrt(x))  
        return (sr * sr) == x  
    return False
```

```
n = 84  
if isPerfectSquare(n):  
    print("True")  
else:  
    print("False")
```

## Q20: Power of a Number

Write a program to find the power of number using recursion

### Code in C

```
#include <stdio.h>  
#include <math.h>  
  
int powRec(int a,int b)  
{  
    if(b==0)  
        return 1;  
    return a*powRec(a,b-1);  
}  
int main()  
{  
    int a = 2,b=3;  
    printf("The number %d to the power %d is %d",a,b,powRec(a,b));  
}
```

## Code in C++

```
#include<bits/stdc++.h>
using namespace std;

//Recursive Function
int power(int base, int x){

    if(x==0) //Base Condition
        return 1;

    return (base*power(base, x-1));
}
//Driver Code
int main(){
    int base = 5, x = 3;
    cout<<"Required Power is "<<power(base, x);
}
```

## Code in Java

```
public class Main
{
    public static void main (String[]args)
    {
        int base = 5, x = 3;
        System.out.println ("Required Power is " + power (base, x));
    }
    //Recursive Function
    static int power (int base, int x)
    {
        if (x == 0)      //Base Condition
            return 1;
        return (base * power (base, x - 1));
    }
}
```

## Code in Python

```
def power(a, b):
    if b != 0:
        return a * power(a, b - 1)
    else:
```

```
return 1
```

```
a = 2
b = 3
print(a, "to the power", b, "is", power(a, b))
```

## Q21: Program to calculate length of the string using recursion

Write a Program to calculate length of the string using recursion

### Code in C

```
#include<stdio.h>
int find_len (char [], int);

int main ()
{
    char str[100]="Let's Learn C Programming";
    int len = 0;

    len = find_len (str, 0);

    printf ("The length of the given string is: %d\n", len);
    return 0;
}

int find_len (char str[], int index){
    static int l = 0;
    if (str[index] == '\0')
        return l;
    else
        l ++;
    find_len (str, index + 1);
}
```

### Code in C++

```
#include <bits/stdc++.h>
```

```

using namespace std;

//Recursive function to calculate the length of the string
int Len(char* str)
{
    if (*str == '\0')
        return 0;
    else
        return 1 + Len(str + 1);
}

/* Driver code */
int main()
{
    char str[] = "PrepInsta";
    cout << Len(str);
    return 0;
}

```

## Code in Java

```

public class Main
{
    //Function to calculate length
    private static int recLength(String str)
    {
        // if we reach at the end of the string
        if (str.equals(""))
            return 0;
        else
            return recLength(str.substring(1)) + 1;
    }

    //Driver program to test the function
    public static void main(String[] args)
    {
        String str = "Prepinsta";
        System.out.println("length of the string "+recLength(str));
    }
}

```

## Code in Python

```
def length(str):
    if str == "":
        return 0
    return 1 + length(str[1:])

str = "Preplnsta"
print("length of", str, "is", length(str))
```

## Q22: Decimal to Octal Conversion

Write a program to convert decimal numbers to octal numbers.

### Code in C

```
#include<stdio.h>

void convert(int num)
{
    // creating an array to store octal equivalent
    int octalArray[32];

    // using i to store octal bit at given array position
    int i = 0;
    while (num > 0) {

        // resultant remainder is stored at given array position
        octalArray[i] = num % 8;
        num = num / 8;
        i++;
    }

    // printing octal array in reverse order
    for (int j = i - 1; j >= 0; j--)
        printf("%d", octalArray[j]);
}

int main()
{
    int n = 148;
    convert(n);
    return 0;
}
```

## Code in C++

```
#include<iostream>
using namespace std;

void convertOctal(int num)
{
    // creating an array to store octal equivalent
    int octalArray[32];

    // using i to store octal bit at given array position
    int i = 0;
    while (num > 0) {

        // resultant remainder is stored at given array position
        octalArray[i] = num % 8;
        num = num / 8;
        i++;
    }

    // printing octal array in reverse order
    for (int j = i - 1; j >= 0; j--)
        cout << octalArray[j];
}

int main()
{
    int n = 148;
    convertOctal(n);
    return 0;
}
```

## Code in Java

```
import java.util.Scanner;
public class Main
{
    public static void main(String args[])
    {
        //scanner class object creation
        Scanner sc = new Scanner(System.in);
        //Number
        int decimal = 148;
        //integer array for storing octal digits
        int octal[] = new int[20];
    }
}
```

```

        int i = 0;
        //writing logic for the conversion
        while(decimal > 0)
        {
            int r = decimal % 8;
            octal[i++] = r;
            decimal = decimal/8;
        }
        //printing result
        System.out.print("Octal number : ");
        for(int j = i-1 ; j >= 0 ; j--)
            System.out.print(octal[j]);
        //closing scanner class(not compulsory, but good practice)
        sc.close();
    }
}

```

### Code in Python

```

decimal = 148
octal = []
while decimal > 0:
    r = decimal % 8
    octal.append(r)
    decimal = decimal // 8
for i in reversed(octal):
    print(i, end="")

```

## Q23: Given a positive integer N, return the Nth row of pascal's triangle

Given a positive integer N, write a program to return the Nth row of pascal's triangle

### Code in C

```

#include <stdio.h>

//Function to print N-th row
void getrow(int N)
{
    int prev = 1;
    printf("%d ", prev);
}

```



```

    for (int i = 1; i <= N; i++) {
        int curr = (prev * (N - i + 1)) / i;
        printf("%d ", curr);
        prev = curr;
    }
}

```

// Driver Program

```

int main()
{

    int N = 5;
    getrow(N);
    return 0;
}

```

## Code in C++

```

#include <bits/stdc++.h>
using namespace std;

//Recursive Function
vector<int> getRow(int index)
{
    vector<int> cur_row;

    // 1st element of every row is 1
    cur_row.push_back(1);

    if (index == 0)
    {
        return cur_row;
    }

    vector<int> prev = getRow(index - 1);

    for(int i = 1; i < prev.size(); i++)
    {
        int curr = prev[i - 1] + prev[i];
        cur_row.push_back(curr);
    }
    cur_row.push_back(1);
}

```

```

    return cur_row;
}

// Driver Code
int main()
{
    int n = 2;
    vector<int> arr = getRow(n);

    for(int i = 0; i < arr.size(); i++)
    {
        if (i == arr.size() - 1)
            cout << arr[i];
        else
            cout << arr[i] << " ";
    }
    return 0;
}

```

## Code in Java

```

import java.util.ArrayList;
public class Main {

    public static ArrayList getRow(int rowIndex)
    {
        ArrayList <Integer> currow = new ArrayList();

        currow.add(1);

        if (rowIndex == 0) {
            return currow;
        }

        ArrayList <Integer> prev = getRow(rowIndex- 1);

        for (int i = 1; i < prev.size(); i++) {

            int curr = prev.get(i - 1) + prev.get(i);
            currow.add(curr);
        }
    }
}

```

```

    }
    currow.add(1);

    return currow;
}

// Driver Program
public static void main(String[] args)
{
    int n = 3;
    ArrayList arr = getRow(n);

    for (int i = 0; i < arr.size(); i++) {
        if (i == arr.size() - 1)
            System.out.print(arr.get(i));
        else
            System.out.print(arr.get(i) + ", ");
    }
}
}

```

## Code in Python

```

def getRow(rowIndex):

    cur_row = []

    cur_row.append(1)

    if rowIndex == 0:
        return cur_row

    prev = getRow(rowIndex - 1)

    for i in range(1, len(prev)):
        curr = prev[i - 1] + prev[i]
        cur_row.append(curr)

    cur_row.append(1)

    return cur_row

```

```

n = 2
arr = getRow(n)

for i in range(len(arr)):

    if i == (len(arr) - 1):
        print(arr[i])
    else:
        print(arr[i], end=" ")

```

## Q24: Decimal to Hexadecimal Conversion

Write a program to convert decimal numbers to hexadecimal numbers.

### Code in C

```

#include<stdio.h>

void convert (int num)
{
    // creating a char array to store hexadecimal equivalent
    char hexa[100];

    // using i to store hexadecimal bit at given array position
    int i = 0;
    while (num != 0)
    {
        int rem = 0;

        rem = num % 16;

        if (rem < 10) { hexa[i] = rem + 48; i++; }

        else
        {
            hexa[i] = rem + 55;
            i++;
        }
        num = num / 16;
    }
    // printing hexadecimal array in reverse order

```

```
printf ("Hexadecimal:");  
    for (int j = i- 1; j >= 0; j--)  
        printf ("%c" , hexa[j]);  
}
```

```
int main ()  
{  
    int decimal = 1457;  
    convert (decimal);  
    return 0;  
}
```

## Code in C++

```
#include<iostream>
```

```
using namespace std;
```

```
void  
getHexadecimal (int decimal)  
{
```

```
    char result[100];
```

```
    int pos = 0;
```

```
while (decimal != 0)  
{
```

```
int rem = 0;
```

```
rem = decimal % 16;
```

```
    if (rem < 10)
```

```
    {
```

```
result[pos] = rem + 48;
```

```
pos++;
```

```

    }
    else {
        result[pos] = rem + 55;
        pos++;
    }
    decimal = decimal / 16;

    } // to get result we need to read the array in opposite fashion
    cout << "Hexadecimal Value: ";
    for (int j = pos - 1; j >= 0; j--)
        cout << result[j];

}
int main()
{
    int decimal;
    cout << "Decimal Value:" ;
    cin >> decimal;
    getHexadecimal(decimal);
    return 0;
}

```

## Code in Java

```

public class Main
{

    public static void main (String[]args)
    {

        int decimal = 1457;

        convert (decimal);

    }
}

```

```

static void convert (int num)
{

    // creating a char array to store hexadecimal equivalent
    char[] hexa = new char[100];

    // using i to store hexadecimal bit at given array position
    int i = 0;

    while (num != 0)

    {

        int rem = 0;

        rem = num % 16;

        if (rem < 10)
        {
            hexa[i] = (char) (rem + 48);
            i++;
        }
        else

        {

            hexa[i] = (char) (rem + 55);

            i++;

        }
        num = num / 16;

    }
    // printing hexadecimal array in reverse order
    System.out.println ("Hexadecimal:");

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

```

```
System.out.print (hexa[j]);
```

```
}
```

```
}
```

## Code in Python

```
def convert(num):
```

```
    hexa = []
```

```
    while num != 0:
```

```
        rem = num % 16
```

```
        if rem < 10:
```

```
            hexa.append(chr(rem + 48))
```

```
        else:
```

```
            hexa.append(chr(rem + 55))
```

```
        num = num // 16
```

```
    hexa.reverse()
```

```
    return "".join(hexa)
```

```
decimal = 2545
```

```
print("Hexadecimal :", convert(decimal))
```

## Q25: Find Largest element in an array

Write a program to find the largest element in an array

## Code in C

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

```
int getLargest(int arr[], int len)
```

```
{
```

```
    // assign first array element as largest
```

```
    int max = arr[0];
```

```
    // linearly search for the largest element
```

```
    for(int i=1; i<len;i++){
```



```

        if(arr[i] >max) max = arr[i];
    }

    return max;
}
int main()
{
    int arr[] = {20, 5, 35, 40, 10, 50, 15};

    // get the length of the array
    int len = sizeof(arr)/sizeof(arr[0]);

    printf("The Largest element is: %d", getLargest(arr, len));
}

```

## Code in C++

```

#include<bits/stdc++.h>
using namespace std;

int main(){

    int arr[]={10, 89, 67, 56, 45, 78};
    int n = sizeof(arr)/sizeof(arr[0]);
    int max_element = INT_MIN;

    for(int i=0; i<n; i++){
        if(arr[i]>max_element)
            max_element = arr[i];
    }

    cout<<max_element;
}

```

## Code in Java

```

import java.util.Scanner;

public class Main
{
    public static void main(String args[])
    {

        int arr[] = {12, 13, 1, 10, 34, 10};
    }
}

```

```

        int max = arr[0];

        for(int i=0; i<arr.length; i++)
        {
            if(max < arr[i])
            {
                max = arr[i];
            }
        }

        System.out.print(max);
    }
}

```

Code in Python

## Q26: Binary to Octal conversion

Write a program to convert binary numbers to octal numbers.

Code in C

```

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

// function to convert binary to octal
void convert(long long num)
{
    int octalDigit = 0, count = 1, i = 0, pos = 0;
    int octalArray[32] = {0};

    while(num != 0)
    {
        int digit = num % 10;

        octalDigit += digit * pow(2, i);
        i++;
        num /= 10;

        // placing current octal sum for 3 pair in array index position
    }
}

```

```

        octalArray[pos] = octalDigit;

        if(count % 3 == 0)
        {
            octalDigit = 0;
            i = 0;
            pos++;
        }
        count++;
    }

    // printing octal array in reverse order
    for (int j = pos; j >= 0; j--)
        printf("%d",octalArray[j]);

}

//main program
int main()
{
    // long used rather than int to store large values
    long long binary;

    printf("Enter binary number: ");
    scanf("%lld", &binary);

    convert(binary);

    return 0;
}

```

## Code in C++

```

#include<iostream>
#include<math.h>
using namespace std;

// function to convert binary to octal
void convert(long long num)
{
    int octalDigit = 0, count = 1, i = 0, pos = 0;
    int octalArray[32] = {0};

    while(num != 0)

```

```

{
    int digit = num % 10;

    octalDigit += digit * pow(2, i);
    i++;
    num /= 10;

    // placing current octal sum for 3 pair in array index position
    octalArray[pos] = octalDigit;

    // whenever we have read next 3 digits
    // setting values to default
    // increasing pos so next values can be placed at next array index
    if(count % 3 == 0)
    {
        octalDigit = 0;
        i = 0;
        pos++;
    }
    count++;
}

// printing octal array in reverse order
for (int j = pos; j >= 0; j--)
    cout<<octalArray[j];

}

//main program
int main()
{
    // long used rather than int to store large values
    long long binary;

    printf("Enter binary number: ");
    scanf("%lld", &binary);

    convert(binary);

    return 0;
}

```

## Code in Java

```
import java.util.Scanner;
```

```

public class Main
{
    public static void main(String args[])
    {
        //scanner class object creation
        Scanner sc = new Scanner(System.in);
        //input from user
        System.out.print("Enter a binary number : ");
        int binary = sc.nextInt();
        //Declaring variable to store decimal number
        int decimal = 0;
        //Declaring variable to use in power
        int n = 0;
        //writing logic for the conversion from binary to decimal
        while(binary > 0)
        {
            int temp = binary%10;
            decimal += temp*Math.pow(2, n);
            binary = binary/10;
            n++;
        }
        int octal[] = new int[20];
        int i = 0;
        //writing logic for the conversion from decimal to octal
        while(decimal > 0)
        {
            int r = decimal % 8;
            octal[i++] = r;
            decimal = decimal / 8;
        }
        //printing result
        System.out.print("Octal number : ");
        for(int j = i-1 ; j >= 0 ; j--)
            System.out.print(octal[j]);
        //closing scanner class(not compulsory, but good practice)
        sc.close();
    }
}

```

## Code in Python

```

def convert(num):
    octalDigit = 0
    count = 1
    i = 0

```

```

pos = 0
octalArray = [0] * 32

while num != 0:
    digit = num % 10
    octalDigit += digit * pow(2, i)
    i += 1
    num //= 10

# placing current octal-sum for 3 pair in array index position
octalArray[pos] = octalDigit

if count % 3 == 0:
    octalDigit = 0
    i = 0
    pos += 1

count += 1

# printing octal array in reverse order
for j in range(pos, -1, -1):
    print(octalArray[j], end="")

binary = 1010
convert(binary)

```

## Q27: Find the Smallest and largest element in an array

Write a program to find the Smallest and largest element in an array

### Code in C

```

#include <stdio.h>

void getSmallLarge(int arr[], int n)
{
    int smallest, largest;

    smallest = largest = arr[0];

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

```

```

        // finding smallest here
        if(arr[i] < smallest) smallest = arr[i]; // finding largest here
        if(arr[i] > largest) largest = arr[i];

    }
    printf("Smallest: %d\n",smallest);
    printf("Largest: %d", largest);

}
int main()
{
    int arr[] = {25, 40, 35, 20, 10, 80};
    int len = sizeof(arr)/sizeof(arr[0]);
    getSmallLarge(arr, len);

}

```

## Code in C++

```

#include<bits/stdc++.h>
using namespace std;
int main(){

    int arr[] = {10, 67, 89, 78, 34, 2, 95};
    int n = sizeof(arr)/sizeof(arr[0]);

    int smallest = INT_MAX, largest = INT_MIN;

    for(int i=0; i<n; i++){ if(smallest > arr[i])
        smallest = arr[i];
        if(largest < arr[i])
            largest = arr[i];
    }

    cout<<smallest<<endl<<largest;
}

```

## Code in Java

```

import java.util.Scanner;

public class Main
{
    public static void main(String args[])

```

```

{

    int arr[] = {12, 13, 1, 10, 34, 10};

    int largest = arr[0], smallest=arr[0];

    for(int i=0; i<arr.length; i++) { if(smallest > arr[i])
        smallest = arr[i];
        if(largest < arr[i])
            largest = arr[i];
    }

    System.out.println(smallest);
    System.out.println(largest);
}
}

```

## Code in Python

```

arr = [10, 89, 9, 56, 4, 80, 8]
mini = arr[0]
maxi = arr[0]

for i in range(len(arr)):
    if arr[i] < mini: mini = arr[i]

    if arr[i] > maxi: maxi = arr[i]

print (mini)
print (maxi)

```

## Q28: Octal to Binary conversion

Write a program to convert octal numbers to binary numbers.

## Code in C

```

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

void convert(int octal)

```



```

{
    int i = 0, decimal = 0;

    //converting octal to decimal
    while (octal!=0)
    {
        int digit = octal % 10;
        decimal += digit * pow(8, i);

        octal /= 10;
        i++;
    }

    printf("Decimal Value: %d\n",decimal);

    long long binary = 0;
    int rem;
    i = 1;

    // converting decimal to binary here
    while(decimal!=0)
    {
        rem = decimal % 2;
        decimal /= 2;
        binary += rem * i;

        // moving to next position ex: units -> tens
        i *= 10;
    }

    printf("Binary Value: %d",binary);
}

int main()
{
    int octal;

    printf("Octal Value: ");
    scanf("%d", &octal);

    convert(octal);

    return 0;
}

```

## Code in C++

```
#include<iostream>
#include<math.h>
using namespace std;

void
convert (int octal)
{

    int i = 0, decimal = 0;

    //converting octal to decimal
    while (octal != 0)

    {

        int digit = octal % 10;

        decimal += digit * pow (8, i);

        octal /= 10;

        i++;

    }

    printf ("Decimal Value: %d\n", decimal);

    long long binary = 0;

    int rem;

    i = 1;

    // converting decimal to binary here
    while (decimal != 0)

    {
```

```

rem = decimal % 2;

decimal /= 2;

binary += rem * i;

    // moving to next position ex: units -> tens
    i *= 10;

}

cout << binary;

}

int
main ()
{

int octal;

cout << "Octal Value: "; cin >> octal;
    convert (octal);
    return 0;
}

```

## Code in Java

```

class Main
{
    public static void main(String args[])
    {
        int octal = 12;

        //Declaring variable to store decimal number
        int decimal = 0;
        //Declaring variable to use in power
        int n = 0;

        //writing logic for the octal to decimal conversion
    }
}

```

```

while(octal > 0)
{
    int temp = octal % 10;
    decimal += temp * Math.pow(8, n);
    octal = octal/10;
    n++;
}

int binary[] = new int[20];
int i = 0;

//writing logic for the decimal to binary conversion
while(decimal > 0)
{
    int r = decimal % 2;
    binary[i++] = r;
    decimal = decimal/2;
}

//printing result
System.out.print("Binary number : ");

for(int j = i-1 ; j >= 0 ; j--)
    System.out.print(binary[j]+"");

}
}

```

## Code in Python

```

def convert(octal):
    i = 0
    decimal = 0
    while octal != 0:
        digit = octal % 10
        decimal += digit * pow(8, i)
        octal //= 10
        i += 1
    print("Decimal Value :", decimal)
    binary = 0
    rem = 0
    i = 1

    while decimal != 0:
        rem = decimal % 2

```

```
    decimal /= 2
    binary += rem * i
    i *= 10
    print("Binary Value :", binary)
```

```
octal = int(input("Octal Value : "))
convert(octal)
```

## Q29: Find Second Smallest Element in an Array

Write a program to find Second Smallest Element in an Array

### Code in C

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

int secSmallest(int arr[], int n)
{
    // assigning first element as smallest temporarily
    int smallest = arr[0];

    // we find the smallest element here
    for (int i=0; i < n; i++){
        if(arr[i] < smallest)
            smallest = arr[i];
    }

    // temporarily assigning largest max value
    int sec_smallest = INT_MAX;

    // finding second smallest here
    for (int i=0; i < n; i++){
        if(arr[i] != smallest && arr[i] < sec_smallest)
            sec_smallest = arr[i];
    }

    return sec_smallest;
}

int main()
{
```

```

    int arr[] = {70, 40, 30, 20, 10, 90};

    // get the length of the array
    int len = sizeof(arr)/sizeof(arr[0]);

    printf("The 2nd smallest : %d",secSmallest(arr, len));
}

```

## Code in C++

```

#include<bits/stdc++.h>
using namespace std;

int secSmallest(int arr[], int n)
{
    // assigning first element as smallest temporarily
    int smallest = arr[0];

    // we find the smallest element here
    for (int i=0; i < n; i++){
        if(arr[i] < smallest)
            smallest = arr[i];
    }

    // temporarily assinging largest max value
    int sec_smallest = INT_MAX;

    // finding second smallest here
    for (int i=0; i < n; i++){
        if(arr[i] != smallest && arr[i] < sec_smallest)
            sec_smallest = arr[i];
    }

    return sec_smallest;
}

int main()
{
    int arr[] = {70, 40, 30, 20, 10, 90};

    int n = sizeof(arr)/sizeof(arr[0]);

    cout<<secSmallest(arr, n);
}

```

## Code in Java

```
import java.util.Scanner;
import java.util.*;

public class Main
{
    static int secSmallest(int arr[], int n)
    {
        // assigning first element as smallest temporarily
        int smallest = arr[0];

        // we find the smallest element here
        for (int i=0; i < n; i++){
            if(arr[i] < smallest)
                smallest = arr[i];
        }

        // temporarily assigning largest max value
        int sec_smallest = Integer.MAX_VALUE;

        // finding second smallest here
        for (int i=0; i < n; i++){
            if(arr[i] != smallest && arr[i] < sec_smallest)
                sec_smallest = arr[i];
        }

        return sec_smallest;
    }

    public static void main(String args[])
    {
        int arr[] = {12, 13, 1, 10, 34, 10};
        int n = arr.length;
        System.out.print(secSmallest(arr, n));
    }
}
```

## Code in Python

```
import math

arr = [10, 13, 17, 11, 34, 21]
```

```

first = math.inf
second = math.inf

for i in range(0, len(arr)):
    if arr[i] < first:
        first = arr[i]

for i in range(0, len(arr)):
    if arr[i] != first and arr[i] < second:
        second = arr[i]

print(second)

```

## Q30: Addition of two fractions

Write a program to find Addition of two fractions.

### Code in C

```

#include<stdio.h>
int main()
{
    //for initialize variables
    int numerator1, denominator1, numerator2, denominator2, x, y, c, gcd_no;

    //To take user input of numerators and denominators
    printf("Enter the numerator for 1st number : ");
    scanf("%d",&numerator1);
    printf("Enter the denominator for 1st number : ");
    scanf("%d",&denominator1);
    printf("Enter the numerator for 2nd number : ");
    scanf("%d",&numerator2);
    printf("Enter the denominator for 2nd number : ");
    scanf("%d",&denominator2);

    //numerator
    x=(numerator1*denominator2)+(denominator1*numerator2);

    //denominator
    y=denominator1*denominator2;

    // Trick part. Reduce it to the simplest form by using gcd.
    for(c=1; c <= x && c <= y; ++c)
    {

```



```

        if(x%c==0 && y%c==0)
            gcd_no = c;
    }

    //To display fraction of given numerators and denominators
    printf("(%d / %d) + (%d / %d) = (%d / %d)", numerator1, denominator1, numerator2,
denominator2, x/gcd_no, y/gcd_no);

    return 0;
}

```

## Code in C++

```

#include<iostream>
using namespace std;

// GCD function
int findGCD(int n1, int n2)
{
    int gcd;
    for(int i=1; i <= n1 && i <= n2; i++)
    {
        if(n1%i==0 && n2%i==0)
            gcd = i;
    }
    return gcd;
}

// Main Program
int main()
{
    int num1,den1;

    //user input first fraction
    cout << "Enter numerator and denominator of first number : "; cin >> num1 >> den1;

    int num2,den2;

    //user input second fraction
    cout << "Enter numerator and denominator of second number: "; cin >> num2 >> den2;

    //finding lcm of the denominators
    int lcm = (den1*den2)/findGCD(den1,den2);

    //finding the sum of the numbers

```

```

int sum=(num1*lcm/den1) + (num2*lcm/den2);

//normalizing numerator and denominator of result
int num3=sum/findGCD(sum,lcm);

lcm=lcm/findGCD(sum,lcm);

//printing output
cout<<num1<<"/"<<den1<< " + "<<num2<<"/"<<den2<< " = "<<num3<<"/"<<lcm;

return 0;
}

```

## Code in Java

```

import java.util.Scanner;
public class Main
{
    public static void main(String[] args)
    {
        //scanner class declaration
        Scanner sc = new Scanner(System.in);
        //input from the user
        System.out.print("Enter numerator for first fraction : ");
        int num1 = sc.nextInt();
        System.out.print("Enter denominator for first fraction : ");
        int den1 = sc.nextInt();
        System.out.print("Enter numerator for second fraction : ");
        int num2 = sc.nextInt();
        System.out.print("Enter denominator for second fraction : ");
        int den2 = sc.nextInt();
        int num, den, x;
        System.out.print("(" + num1 + " / " + den1 + ") + (" + num2 + " / " + den2 + ") = ");
        //logic for calculating sum of two fractions
        if(den1 == den2)
        {
            num = num1 + num2 ;
            den = den1 ;
        }
        else{
            num = (num1*den2) + (num2*den1);
            den = den1 * den2;
        }
        if(num > den)
            x = num;
    }
}

```

```

else
    x = den;
for(int i = 1 ; i <= x ; i++)
{
    if(num%i == 0 && den%i == 0)
    {
        num = num/i;
        den = den/i;
    }
}
//logic for getting simplified fraction
int n = 1;
int p = num;
int q = den;
if( num != den)
{
    while(n != 0)
    {
        //storing remainder
        n = num % den;
        if(n != 0)
        {
            num = den;
            den = n;
        }
    }
}
System.out.println("(" + p/den + " / " + q/den + ")");
//closing scanner class(not compulsory, but good practice)
sc.close();
}
}

```

## Code in Python

```

def findGCD(n1, n2):
    gcd = 0
    for i in range(1, int(min(n1, n2)) + 1):
        if n1 % i == 0 and n2 % i == 0:
            gcd = i
    return gcd

```

# input first fraction

```
num1, den1 = map(int, list(input("Enter numerator and denominator of first number : ").split(" ")))
```

```

# input first fraction
num2, den2 = map(int, list(input("Enter numerator and denominator of second number: ").split("
")))

lcm = (den1 * den2) // findGCD(den1, den2)

sum = (num1 * lcm // den1) + (num2 * lcm // den2)

num3 = sum // findGCD(sum, lcm)

lcm = lcm // findGCD(sum, lcm)

print(num1, "/", den1, " + ", num2, "/", den2, " = ", num3, "/", lcm)

```

### Q31: Calculate the sum of elements in an array

Write a program to calculate the sum of elements in an array

#### Code in C

```

#include <stdio.h>

int calcSum(int arr[], int len){
    int sum = 0;

    for(int i = 0; i < len; i++)
        sum = sum + arr[i];

    return sum;
}

int main()
{
    int arr[] = {1, 5, 2, 10, 4, 8};

    // get the length of the array
    int len = sizeof(arr)/sizeof(arr[0]);

    printf("Sum: %d",calcSum(arr, len));
}

```

#### Code in C++

```

#include<bits/stdc++.h>

```

```

using namespace std;

int main(){

    int arr[] = {10, 20, 30, 50, 89};

    int n = sizeof(arr)/sizeof(arr[0]);

    int sum =0;

    for(int i=0; i<n; i++){
        sum += arr[i];
    }

    cout<<sum;
}

```

## Code in Java

```

import java.util.Scanner;

public class Main
{
    public static void main(String args[])
    {

        int arr[] = {12, 13, 1, 10, 34, 10};

        int sum = 0;

        for(int i=0; i<arr.length; i++)
        {
            sum = sum + arr[i];
        }

        System.out.print(sum);
    }
}

```

## Code in Python

```

arr = [10, 89, 9, 56, 4, 80, 8]
Sum = 0

for i in range(len(arr)):

```

```
Sum = Sum + arr[i]
print (Sum)
```

## Q32: Reverse an Array

Write a program to reverse an Array

### Code in C

```
#include <stdio.h>

void printReverse(int arr[], int len){

    for(int i = len - 1; i >= 0; i--){
        printf("%d ", arr[i]);
    }

int main()
{
    int arr[] = {10, 20, 30, 40, 50, 60};

    int len = sizeof(arr)/sizeof(arr[0]);

    printf("Array in Reverse:\n");
    printReverse(arr, len);

    return 0;
}
```

### Code in C++

```
#include<bits/stdc++.h>
using namespace std;

int main(){

    int arr[] = {10, 20, 30, 40, 50};
    int n = sizeof(arr)/sizeof(arr[0]);

    for(int i=n-1; i>=0; i--){
        cout<<arr[i]<<" ";
    }
}
```

## Code in Java

```
import java.util.Scanner;

public class Main
{
    public static void main(String args[])
    {

        int arr[] = {10, 20, 30, 40, 50};

        int n=arr.length;
        for(int i=n-1; i>=0; i--)
            System.out.print(arr[i]+" ");
    }
}
```

## Code in Python

```
def reverseList(A, start, end):
    while start < end:
        A[start], A[end] = A[end], A[start]
        start += 1
        end -= 1
# Driver function to test above function
A = [10, 20, 30, 40, 50]
reverseList(A, 0, 4)
print(A)
```

## Q33: Maximum number of handshakes

Write a program to find maximum number of handshakes.

## Code in C

```
#include<stdio.h>

int main()
{

    //fill the code
    int num = 10;
```

```

int total = num * (num-1) / 2; // Combination nC2

printf("For %d people there will be %d handshakes", num, total);

return 0;

}

```

## Code in C++

```

#include<iostream>
using namespace std;

int main()
{

    //fill the code
    int num = 9;

    int total = num * (num-1) / 2; // Combination nC2

    cout<<"For "<<num<<" people there will "<<total<<" handshakes";

    return 0;

}

```

## Code in Java

```

public class Main
{
    public static void main(String[] args) {
        //fill the code
        int num = 10;

        int total = num * (num-1) / 2; // Combination nC2

        System.out.println("For "+ num +" people there will be " +total+" handshakes");
    }
}

```

## Code in Python

```

N = 30

```



```
# formula
no_of_handshakes = int(N * ((N - 1) / 2))

print('Maximum number of handshakes possible for', N, 'pople are', no_of_handshakes)
```

## Q34: Sort the elements of an array

Write a program to sort the elements of an array

### Code in C

```
#include <stdio.h>

void swap(int *xp, int *yp)
{
    int temp = *xp;
    *xp = *yp;
    *yp = temp;
}

void selectionSort(int array[], int size)
{
    int i, j, min_idx;

    // Loop to iterate on array
    for (i = 0; i < size-1; i++)
    {
        // Here we try to find the min element in array
        min_idx = i;
        for (j = i+1; j < size; j++)
        {
            if (array[j] < array[min_idx])
                min_idx = j;
        }
        // Here we interchange the min element with first one
        swap(&array[min_idx], &array[i]);
    }
}

/* Display function to print values */
void display(int array[], int size)
{
```

```

        int i;
        for (i=0; i < size; i++)
        {
            printf("%d ",array[i]);
        }
        printf("\n");
    }

// The main function to drive other functions
int main()
{
    int array[] = {50, 30, 10, 90, 80, 20, 40, 70};
    int size = sizeof(array)/sizeof(array[0]);

    selectionSort(array, size);

    display(array, size);

    return 0;
}

```

## Code in C++

```

#include<bits/stdc++.h>
using namespace std;

void sorted(int arr[], int n){

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

    for(int i=0; i<n; i++)
        cout<<arr[i]<<" ";
}

int main(){

    int arr[] = {10, 89, 67, 45, 83, 9, 12};
    int n = sizeof(arr)/sizeof(arr[0]);
}

```

```
sorted(arr, n);  
}
```

## Code in Java

```
public class Main {  
    public static void main(String[] args) {  
  
        //Initialize array  
        int [] arr = new int [] {10, 40, 30, 20};  
        int temp = 0;  
  
        //Sort the array in ascending order  
        for (int i = 0; i < arr.length; i++) {  
            for (int j = i+1; j < arr.length; j++) { if(arr[i] > arr[j]) {  
                temp = arr[i];  
                arr[i] = arr[j];  
                arr[j] = temp;  
            }  
        }  
    }  
  
    //Displaying elements of array after sorting  
    for (int i = 0; i < arr.length; i++) {  
        System.out.print(arr[i] + " ");  
    }  
}
```

## Code in Python

```
# List of Integers  
numbers = [10, 30, 40, 20]  
  
# Sorting list of Integers  
numbers.sort()  
  
print(numbers)
```

## Q35: Replace all 0's with 1 in a given integer

Write a program to replace all 0's with 1 in a given integer.

## Code in C

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

```
//main program
```

```
int main()
```

```
{
```

```
    int num,num2=0;
```

```
    printf("Enter number: ");
```

```
    //user input
```

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

```
    //checking for 0 input
```

```
    if(num == 0)
```

```
        num2=1;
```

```
    //converting 0 to 1
```

```
    while(num>0)
```

```
{
```

```
    int rem = num%10;
```

```
    if(rem == 0)
```

```
        rem = 1;
```

```
    num = num/10;
```

```
    num2=num2*10+rem;
```

```
}
```

```
num = 0 ; // Store the reverse of num2
```

```
while(num2>0){
```

```

    int r = num2%10;

    num= num*10 + r;

    num2 /= 10;

}

//converted number

printf("Converted number is: %d" ,num);

return 0;

}

```

## Code in C++

```

#include<bits/stdc++.h>
using namespace std;

//main program

int main()

{

    int num,num2=0;

    cout<<"Enter number: "; //user input
    cin>>num;

    //checking for 0 input

    if(num == 0)

        num2=1;

    //converting 0 to 1

    while(num>0)

    {

```

```

        int rem = num%10;

        if(rem == 0)

            rem = 1;

        num = num/10;

        num2=num2*10+rem;

    }

    num = 0 ; // Store the reverse of num2

    while(num2>0){

        int r = num2%10;

        num= num*10 + r;

        num2 /= 10;

    }

    //converted number

    cout<<"Converted number is: "<<num;

    return 0;

}

```

## Code in Java

```

import java.util.Scanner;
public class Main
{
    public static void main(String[] args)
    {
        //scanner class declaration
        Scanner sc = new Scanner(System.in);
        //input from the user
        System.out.print("Enter the number : ");
        int number = sc.nextInt();
        //convert the number to string and then calculate its length
    }
}

```

```

        String str = Integer.toString(number);
        int len = str.length();
        String str1 = "";
        //use the logic to replace all 0's with 1 in a given integer
        for(int i = 0 ; i < len ; i++)
        {
            if(str.charAt(i) == '0')
                str1 = str1 + '1';
            else
                str1 = str1 + str.charAt(i);
        }
        System.out.println("Converted number is: "+str1);
        //closing scanner class(not compulsory, but good practice)
        sc.close();
    }
}

```

## Code in Python

```

num = int(input("Enter number : "))
num2 = 0

if num == 0:
    num2 = 1

while num > 0:
    rem = num % 10
    if rem == 0:
        rem = 1
    num = num//10
    num2 = num2 * 10 + rem

num = 0
while num2 > 0:
    r = num2 % 10
    num = num * 10 + r
    num2 //= 10

print("Converted number is:", num)

```

## Q36: Finding the frequency of elements in an array

Write a program to find the frequency of elements in an array

## Code in C

```
#include<stdio.h>

// Main function to run the program
int main()
{
    int arr[] = {10, 30, 10, 20, 10, 20, 30, 10};
    int n = sizeof(arr)/sizeof(arr[0]);

    int visited[n];

    for(int i=0; i<n; i++){

        if(visited[i]==0){
            int count = 1;
            for(int j=i+1; j<n; j++){
                if(arr[i]==arr[j]){
                    count++;
                    visited[j]=1;
                }
            }

            printf("%d occurs %d times\n", arr[i], count);
        }
    }

    return 0;
}
```

## Code in C++

```
#include <bits/stdc++.h>
using namespace std;

// Main function to run the program
int main()
{
    int arr[] = {10, 30, 10, 20, 10, 20, 30, 10};
    int n = sizeof(arr)/sizeof(arr[0]);

    int visited[n];

    for(int i=0; i<n; i++){
```



```

        if(visited[j]!=1){
            int count = 1;
            for(int j=i+1; j<n; j++){
                if(arr[i]==arr[j]){
                    count++;
                    visited[j]=1;
                }
            }

            cout<<arr[i]<<" occurs at "<<count<<" times "<<endl;
        }
    }

    return 0;
}

```

## Code in Java

```

import java.util.Arrays;
class Main
{
    public static void countFreq(int arr[], int n)
    {
        boolean visited[] = new boolean[n];
        Arrays.fill(visited, false);

        // Traverse through array elements and
        // count frequencies
        for (int i = 0; i < n; i++) {

            // Skip this element if already processed
            if (visited[i] == true)
                continue;

            // Count frequency
            int count = 1;
            for (int j = i + 1; j < n; j++) {
                if (arr[i] == arr[j]) {
                    visited[j] = true;
                    count++;
                }
            }
            System.out.println(arr[i] + " occurs " + count + " times ");
        }
    }
}

```

```
// Driver code
public static void main(String []args)
{
int arr[] = new int[]{10, 30, 10, 20, 10, 20, 30, 10};
int n = arr.length;
countFreq(arr, n);
}
}
```

## Code in Python

```
def countFreq(arr, n):

    # Mark all array elements as not visited
    visited = [False for i in range(n)]

    # Traverse through array elements
    # and count frequencies
    for i in range(n):

        # Skip this element if already
        # processed
        if (visited[i] == True):
            continue

        # Count frequency
        count = 1
        for j in range(i + 1, n, 1):
            if (arr[i] == arr[j]):
                visited[j] = True
                count += 1

        print(arr[i], count)

# Driver Code
arr = [10, 30, 10, 20, 10, 20, 30, 10]
n = len(arr)
countFreq(arr, n)
```

## Q37: Can a number be expressed as a sum of two prime numbers

Write a program to check whether a number can be expressed as a sum of two prime numbers or not.

### Code in C

```
#include<stdio.h>
int sum_of_two_primes(int n);
int main()
{
    int n, i;

    printf("Insert the num: ");
    scanf("%d", &n);
    int flag = 0;
    for(i = 2; i <= n/2; ++i)
    {
        // Condition for i to be prime
        if(sum_of_two_primes(i) == 1)
        {
            if(sum_of_two_primes(n-i) == 1)
            {
                printf("%d can be expressed as the sum of %d and %d", n, i, n-i);
                flag = 1;
            }
        }
    }

    if(flag == 0)
        printf("%d cannot be expressed as the sum of two primes\n", n);
    return 0;
}

int sum_of_two_primes(int n)
{
    int i, isPrime = 1;
    for(i = 2; i <= n/2; ++i)
    {
        if(n % i == 0)
        {
            isPrime = 0;
        }
    }
}
```

```

        break;
    }
}
return isPrime;
}

```

## Code in C++

```

#include <iostream>
using namespace std;
bool prime_or_not(int a);
int main() {
    int a, i;
    bool flag = false;
    cout << "Enter a positive integer: ";
    cin >> a;
    for(i = 2; i <= a/2; ++i) {
        if (prime_or_not(i)) {
            if (prime_or_not(a - i)) {
                cout << a << " = " << i << " + " << a-i << endl;
                flag = true;
            }
        }
    }
    if (!flag)
        cout << a << " can't be expressed as sum of two prime numbers.";
    return 0;
}
bool prime_or_not(int a) {
    int i;
    bool is_prime = true;
    if (a == 0 || a == 1) {
        is_prime = false;
    }
    for(i = 2; i <= a/2; ++i) {
        if(a % i == 0) {
            is_prime = false;
            break;
        }
    }
    return is_prime;
}

```

## Code in Java

```
import java.util.Scanner;
public class Main
{
    public static void main(String[] args)
    {
        //scanner class declaration
        Scanner sc = new Scanner(System.in);
        //input from user
        System.out.print("Enter a number : ");
        int number = sc.nextInt();
        int x = 0;
        for(int i = 2 ; i <= number/2 ; i++)
        {
            if(prime_or_not(i) == 1)
            {
                if(prime_or_not(number-i) == 1)
                {
                    System.out.println(number+ " = "+i+" + "+(number-i));
                    x = 1;
                }
            }
        }
        if(x == 0)
            System.out.println(+number+" cannot be expressed as a sum of two
prime numbers");
    }
    //function for checking number is prime or not
    public static int prime_or_not(int n)
    {
        int c = 1;
        for(int i = 2 ; i < n ; i++)
        {
            if(n % i == 0)
            {
                c = 0;
                break;
            }
        }
        return c;
    }
}
```

## Code in Python

```
Number = int(input('Enter the Number : '))
# initialize an array
arr = []
# find prime numbers
for i in range(2, Number):
    flag = 0
    for j in range(2, i):
        if i % j == 0:
            flag = 1
    # append prime numbers to array
    if flag == 0:
        arr.append(i)
# possible combinations
flag = 0
for i in range(len(arr)):
    for j in range(i + 1, len(arr)):
        # if condition is True Print numbers
        if arr[i] + arr[j] == Number:
            flag = 1
            print(str(arr[i]) + " and " + str(arr[j]) + ' are prime numbers when added gives ' +
str(Number))
            break
if flag == 0:
    print('No Prime numbers can give sum of ' + str(Number))
```

## Q38: Sorting elements of an array by frequency

Write a program for sorting elements of an array by frequency

### Code in C

```
#include<stdio.h>
#define MAX 256
int main ()
{
    int a[]={1, 2, 1, 1, 2, 3, 3, 3, 3, 0};
    int n = sizeof(a)/sizeof(a[0]);
    int arr[MAX][2], brr[MAX][2];
    int k = 0, temp, count;
    for (int i = 0; i < n; i++)
    {
        arr[i][0] = a[i];
```

```

arr[i][1] = 0;
}
// Unique elements and its frequency are stored in another array
for (int i = 0; i < n; i++)
{
    if (arr[i][1])
        continue;
    count = 1;
    for (int j = i + 1; j < n; j++)
    {
        if (arr[i][0] == arr[j][0])
        {
            arr[j][1] = 1;
            count++;
        }
    }
    brr[k][0] = arr[i][0];
    brr[k][1] = count;
    k++;
}
n = k;

```

```

//Store the array and its frequency in sorted form
for (int i = 0; i < n - 1; i++)
{
    temp = brr[i][1];
    for (int j = i + 1; j < n; j++)
    {
        if (temp < brr[j][1])
        {
            temp = brr[j][1];
            brr[j][1] = brr[i][1];
            brr[i][1] = temp;

            temp = brr[j][0];
            brr[j][0] = brr[i][0];
            brr[i][0] = temp;
        }
    }
}
for (int i = 0; i < n; i++)
{
    while (brr[i][1] != 0)
    {

```

```

        printf (" %d ", brr[i][0]);
        brr[i][1]--;
    }
}
return 0;
}

```

## Code in C++

```

#include <bits/stdc++.h>
using namespace std;
#define MAX 256
int main ()
{
    int a[]={10, 20, 10, 10, 20, 30, 30, 30, 30, 0};
    int n = sizeof(a)/sizeof(a[0]);
    int arr[MAX][2], brr[MAX][2];
    int k = 0, temp, count;

    for (int i = 0; i < n; i++){
        arr[i][0] = a[i];
        arr[i][1] = 0;
    }

    // Unique elements and its frequency are stored in another array
    for (int i = 0; i < n; i++){
        if (arr[i][1])
            continue;
        count = 1;
        for (int j = i + 1; j < n; j++){
            if (arr[i][0] == arr[j][0]){
                arr[j][1] = 1;
                count++;
            }
        }
        brr[k][0] = arr[i][0];
        brr[k][1] = count;
        k++;
    }
    n = k;

    //Store the array and its frequency in sorted form
    for (int i = 0; i < n - 1; i++)
    {
        temp = brr[i][1];

```



```

    for (int j = i + 1; j < n; j++)
    {
        if (temp < brr[j][1])
        {
            temp = brr[j][1];
            brr[j][1] = brr[i][1];
            brr[i][1] = temp;

            temp = brr[j][0];
            brr[j][0] = brr[i][0];
            brr[i][0] = temp;
        }
    }
}
for (int i = 0; i < n; i++)
{
    while (brr[i][1] != 0)
    {
        cout<< brr[i][0] <<" ";
        brr[i][1]--;
    }
}
return 0;
}

```

## Code in Java

```

import java.util.*;
public class Main {
    static Integer[] arr = {10, 20, 10, 10, 20, 30, 30, 30, 30, 0};

    public static void sortBasedOnFrequencyAndValue(List<Integer> list)
    {
        int n = arr.length;
        final HashMap<Integer, Integer> mapCount = new HashMap<Integer, Integer>();
        final HashMap<Integer, Integer> mapIndex = new HashMap<Integer, Integer>();

        for (int i = 0; i < n; i++) {
            if (mapCount.containsKey(arr[i])) {
                mapCount.put(arr[i], mapCount.get(arr[i]) + 1);
            }
            else {
                mapCount.put(arr[i], 1); // Map to capture Count of elements
                mapIndex.put(arr[i], i); // Map to capture 1st occurrence of elements
            }
        }
    }
}

```

```

    }
    }
    Collections.sort(list, new Comparator<Integer>(){
    public int compare(Integer n1, Integer n2)
        {
            int freq1 = mapCount.get(n1);
            int freq2 = mapCount.get(n2);
            if (freq1 != freq2) {
                return freq2 - freq1;
            }
            else {
                return mapIndex.get(n1) - mapIndex.get(n2);
            }
        }
    });
    System.out.println(list);
}

public static void main(String[] args) {
    List<Integer> list = Arrays.asList(arr);
    sortBasedOnFrequencyAndValue(list);
}
}

```

## Code in Python

```

from collections import Counter
ini_list = [10, 20, 30, 40, 40, 50, 50, 50]

# sorting on basis of frequency of elements
result = [item for items, c in Counter(ini_list).most_common() for item in [items] * c]

# printing final result
print(str(result))

```

## Q39: Calculate the area of a circle

Write a Program to calculate the area of a circle.

## Code in C

```

#include <stdio.h>

int main(){

```

```

float r =3, pi=3.14, area;

area=pi*r*r;
printf("Area of circle is %.2f", area);
}

```

## Code in C++

```

#include<bits/stdc++.h>
using namespace std;

int main(){
    float rad = 3, area;

    area=(3.14*rad*rad);
    cout<<"Area of circle is "<<area;
}

```

## Code in Java

```

import java.util.Scanner;
public class Main
{
    public static void main(String[] args)
    {

        double radius = 7;

        //formula for area of a circle
        double area = 3.14 * radius * radius;
        System.out.println(area);

    }
}

```

## Code in Python

```

from math import pi
r=3.00
area=pi*r*r
print("The area of circle is",end=" ")
print(area)

```

## Q40: Find the prime numbers between 1 to 100

Write a program to find the prime numbers between 0 to 100.

### Code in C

```
#include <stdio.h>

int checkPrime(int num)
{
    // 0, 1 and negative numbers are not prime
    if(num < 2){
        return 0;
    }
    else{
        int x = num/2;
        for(int i = 2; i <=x; i++)
        {
            if(num % i == 0)
            {
                return 0;
            }
        }
    }
    // the number would be prime if we reach here
    return 1;
}

int main()
{
    int a = 1, b = 100;

    for(int i=a; i <= b; i++){
        if(checkPrime(i))
            printf("%d ",i);
    }

    return 0;
}
```

### Code in C++

```
#include <iostream>
using namespace std;
```

```

int checkPrime(int num)
{
    if(num < 2){
        return 0;
    }
    else{
        int x = num/2;
        for(int i = 2; i < x; i++)
        {
            if(num % i == 0)
            {
                return 0;
            }
        }
    }

    return 1;
}

int main()
{
    int a = 1, b = 100;

    for(int i=a; i <= b; i++){
        if(checkPrime(i))
            cout<<i<<" ";
    }

    return 0;
}

```

## Code in Java

```

public class Main
{

    public static void main(String[] args) {
        int a=1,b=100;
        for(int i=a;i<=b;i++){
            if(checkPrime(i)){
                System.out.print(i+" ");
            }
        }
    }

    public static boolean checkPrime(int num){

```

```

// 0, 1 and negative numbers are not prime
if(num<2){
    return false;
}
else{

    int x= num/2;
    for(int i=2;i<x;i++){
        if(num%i==0){
            return false;
        }
    }
}
// the number would be prime if we reach here
return true;
}
}

```

## Code in Python

```
def checkPrime(num):
```

```

    # 0, 1 and negative numbers are not prime
    if num < 2:
        return 0
    else:

```

```

        x = num // 2
        for j in range(2, x + 1):
            if num % j == 0:
                return 0

```

```

    # the number would be prime if we reach here
    return 1

```

```

a, b = 1, 100
for i in range(a, b + 1):
    if checkPrime(i):
        print(i, end=" ")

```

## Q41: Finding the Longest Palindrome in an Array

Write a program for finding the Longest Palindrome in an Array

### Code in C

```
#include <stdio.h>
#include <limits.h>
int ispalindrome(int n){
    int rev=0, temp = n;

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

    if(n==rev)
        return 1;

    return 0;
}

int main(){
    int arr[] = {1, 121, 55551, 545545, 10111, 90};
    int n = sizeof(arr)/sizeof(arr[0]);
    int res = INT_MIN;

    for(int i=0; i<n; i++){

        if(ispalindrome(arr[i]) && res<arr[i])
            res = arr[i];

    }

    if(res==INT_MIN)
        res = -1;

    printf("%d ",res);
}
```

### Code in C++

```
#include<bits/stdc++.h>
```

```

using namespace std;

int ispalindrome(int n){
    int rev=0, temp = n;

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

    if(n==rev)
        return 1;

    return 0;
}

int main(){
    int arr[] = {1, 121, 55551, 545545, 10111, 90};
    int n = sizeof(arr)/sizeof(arr[0]);
    int res = INT_MIN;

    for(int i=0; i<n; i++){
        if(ispalindrome(arr[i]) && res<arr[i])
            res = arr[i];
    }

    if(res==INT_MIN)
        res = -1;

    cout<<res;
}

```

## Code in Java

```

import java.util.*;

class Main
{
    // Function to check if n is palindrome
    static boolean isPalindrome(int n)
    {
        // Find the appropriate divisor
        // to extract the leading digit
        int divisor = 1;
    }
}

```



```

while (n / divisor >= 10)
    divisor *= 10;

while (n != 0) {
    int x = n / divisor;
    int y = n % 10;

    // If first and last digits are
    // not same then return false
    if (x != y)
        return false;

    // Removing the leading and trailing
    // digits from the number
    n = (n % divisor) / 10;

    // Reducing divisor by a factor
    // of 2 as 2 digits are dropped
    divisor = divisor / 100;
}
return true;
}

// Function to find the largest palindromic number
static int largestPalindrome(int []A, int n)
{
    int res = -1;

    for (int i = 0; i < n; i++) { // If a palindrome larger than the currentMax is found
        if (A[i] > res && isPalindrome(A[i]))
            res = A[i];
    }

    // Return the largest palindromic number from the array
    return res;
}

// Driver program
public static void main(String []args)
{
    int []A = { 121, 2322, 54545, 999990 };
    int n = A.length;

    // print required answer

```

```

        System.out.println(largestPalindrome(A, n));
    }
}

```

## Code in Python

```

def isPalindrome(n):

    divisor = 1
    while (int(n / divisor) >= 10):
        divisor *= 10

    while (n != 0):
        leading = int(n / divisor)
        trailing = n % 10

        if (leading != trailing):
            return False

        n = int((n % divisor) / 10)

        divisor = int(divisor / 100)
    return True

# Function to find the largest palindromic element
def largestPalindrome(arr, n):
    currentMax = -1

    for i in range(0, n, 1):
        if (arr[i] > currentMax and isPalindrome(arr[i])):
            currentMax = arr[i]

    return currentMax

# Driver Code

arr = [1, 232, 5545455, 909090, 161]
n = len(arr)

# print required answer
print(largestPalindrome(arr, n))

```

## Q42: Calculate the number of digits in an integer

Write a program to calculate the number of digits in an integer.

### Code in C

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

int main(){
    int n = 20901;
    int count = 0;

    while(n>0){
        count++;
        n = n/10;
    }
    printf("No. of digits = %d", count);
}
```

### Code in C++

```
#include<bits/stdc++.h>
using namespace std;

int main(){
    int num = 20901;
    int digit = 0;

    while(num>0){
        digit++;
        num = num/10;
    }
    cout<<"No. of digits = "<< digit;
}
```

### Code in Java

```
import java.util.Scanner;
class Main{
    public static void main(String[] args)
    {
        int number = 12345;
```

```

        //declare a variable to count number of digits
        int digit = 0;
        while(number != 0)
        {
            //pick last digit of the number and count one by one
            int pick_last = number % 10;
            digit++;
            number = number / 10;
        }

        //display number of digits
        System.out.print("Number of Digits = "+digit);

    }
}

```

## Code in Python

```

def countDigit(n):
    digit = 0
    while n != 0:
        n //= 10
        digit += 1
    return digit

```

```

# Driver Code
n = 78673
print("Number of digits : % d" % (countDigit(n)))

```

## Q43: Convert digit/number to words

Write a program to convert digits to words.

## Code in C

```

#include <stdio.h>
#include <string.h>
#include <stdlib.h>
void convert_to_words(char* num)
{
    int len = strlen(num);

```

```

/* Base cases */
if (len == 0) {
    fprintf(stderr, "empty string\n");
    return;
}
if (len > 4) {
    fprintf(stderr,
        "Length more than 4 is not supported\n");
    return;
}

char* single_digits[] = { "zero", "one", "two", "three", "four", "five", "six", "seven", "eight",
    "nine" };

char* two_digits[] = { "", "ten", "eleven", "twelve", "thirteen", "fourteen", "fifteen", "sixteen",
    "seventeen", "eighteen", "nineteen" };

char* tens_multiple[] = { "", "", "twenty", "thirty", "forty", "fifty", "sixty", "seventy", "eighty",
    "ninety" };

char* tens_power[] = { "hundred", "thousand" };

printf("\n%s: ", num);

if (len == 1) {
    printf("%s\n", single_digits[*num - '0']);
    return;
}

while (*num != '\0') {

    if (len >= 3) {
        if (*num - '0' != 0) {
            printf("%s ", single_digits[*num - '0']);
            printf("%s ", tens_power[len - 3]);
        }
        --len;
    }

    else {
        if (*num == '1') {
            int sum = *num - '0' + *(num + 1) - '0';
            printf("%s\n", two_digits[sum]);
            return;
        }
    }
}

```

```

    }

    else if (*num == '2' && *(num + 1) == '0') {
        printf("twenty\n");
        return;
    }

    else {
        int i = *num - '0';
        printf("%s ", i ? tens_multiple[i] : "");
        ++num;
        if (*num != '0')
            printf("%s ",
                single_digits[*num - '0']);
    }
}
++num;
}
}

int main(void)
{
    convert_to_words("9459");
    return 0;
}

```

## Code in C++

```

#include<bits/stdc++.h>
using namespace std;

void
numToWords (string num)
{

    int length_of_string = num.size ();

    if (length_of_string == 0)
    {

        cout << "String is Empty";
        return;
    }
}

```

```

    }
    if (length_of_string > 4)
    {

cout << "Please enter the string with supported length";

return;

}

string ones_digits[] =
{
    "zero", "one", "two", "three", "four", "five", "six", "seven", "eight",
    "nine"};

string tens_digits[] =
{
    "", "ten", "eleven", "twelve", "thirteen", "fourteen", "fifteen",
    "sixteen", "seventeen", "eighteen", "nineteen"};

string multiple_of_ten[] =
{
    "", "", "twenty", "thirty", "forty", "fifty", "sixty", "seventy",
    "eighty", "ninety"};

string power_of_ten[] =
{
    "hundred", "thousand"};

cout << num << ":\n";

if (length_of_string == 1)
{

cout << ones_digits[num[0] - '0'];

    //return;
}

```

```

int x = 0;

while (x < num.size ())
{
    if (length_of_string >= 3)
    {

if (num[x] - 48 != 0)
    {

cout << ones_digits[num[x] - 48] << "\n";

cout << power_of_ten[length_of_string - 3] << "\n";

length_of_string--;

}

}

        else
        {

if (num[x] - 48 == 1)
    {

int sum = (num[x] - 48 + num[x] - 48);

cout << tens_digits[sum];

                // return;

            }

            else if (num[x] - 48 == 2 and num[x + 1] - 48 == 0)
            {

cout << "twenty";        //
                return;

            }
            else

```



```

        {
            int i = num[x] - 48;
            if (i > 0)
            {

cout << multiple_of_ten[i] << " ";

            }

            else
            {

cout << " ";

            }

            x += 1;

            if (num[x] - 48 != 0)
            {

cout << ones_digits[num[x] - 48];

            }

        }

    }

    x++;

}

}

int
main ()
{

```

```
numToWords ("1121");
```

```
return 0;
```

```
}
```

## Code in Java

```
class Main {
```

```
    static void convert_to_words(char[] num)
    {
```

```
        int len = num.length;
```

```
        // Base cases
```

```
        if (len == 0) {
            System.out.println("empty string");
            return;
        }
```

```
        if (len > 4) {
            System.out.println(
                "Length more than 4 is not supported");
            return;
        }
```

```
        String[] single_digits = new String[] {
            "zero", "one", "two", "three", "four",
            "five", "six", "seven", "eight", "nine"
        };
    }
```

```
        String[] two_digits = new String[] {
            "", "ten", "eleven", "twelve",
            "thirteen", "fourteen", "fifteen", "sixteen",
            "seventeen", "eighteen", "nineteen"
        };
    }
```

```
        String[] tens_multiple = new String[] {
            "", "", "twenty", "thirty", "forty",
            "fifty", "sixty", "seventy", "eighty", "ninety"
        };
    }
```

```
String[] tens_power = new String[] { "hundred", "thousand" };
```

```
System.out.print(String.valueOf(num) + ": ");
```

```
if (len == 1) {  
    System.out.println(single_digits[num[0] - '0']);  
    return;  
}
```

```
int x = 0;  
while (x < num.length) { if (len >= 3) {  
    if (num[x] - '0' != 0) {  
        System.out.print(single_digits[num[x] - '0'] + " ");  
        System.out.print(tens_power[len - 3] + " ");  
  
        }  
        --len;  
    }  
}
```

```
else {  
  
    if (num[x] - '0' == 1) {  
        int sum  
        = num[x] - '0' + num[x + 1] - '0';  
        System.out.println(two_digits[sum]);  
        return;  
    }  
  
    else if (num[x] - '0' == 2  
        && num[x + 1] - '0' == 0) {  
        System.out.println("twenty");  
        return;  
    }  
  
    else {  
        int i = (num[x] - '0');  
        if (i > 0)  
            System.out.print(tens_multiple[i] + " ");  
        else  
            System.out.print("");  
        ++x;  
        if (num[x] - '0' != 0)  
            System.out.println(single_digits[num[x] - '0']);  
    }  
}
```

```

        }
        ++x;
    }
}

// Driver Code
public static void main(String[] args)
{
    convert_to_words("1121".toCharArray());
}
}

```

## Code in Python

```

def convert_to_words(num):

    l = len(num)

    # Base cases
    if (l == 0):
        print("empty string")
        return

    if (l > 4):
        print("Length more than 4 is not supported")
        return

    single_digits = ["zero", "one", "two", "three", "four", "five", "six", "seven", "eight", "nine"]

    two_digits = ["", "ten", "eleven", "twelve", "thirteen", "fourteen", "fifteen", "sixteen",
"seventeen", "eighteen", "nineteen"]

    tens_multiple = ["", "", "twenty", "thirty", "forty", "fifty", "sixty", "seventy", "eighty", "ninety"]

    tens_power = ["hundred", "thousand"]

    print(num, ":", end=" ")

    if (l == 1):
        print(single_digits[ord(num[0]) - 48])
        return

    x = 0
    while (x < len(num)):
        if (l >= 3):

```

```

    if (ord(num[x]) - 48 != 0):
        print(single_digits[ord(num[x]) - 48],
              end=" ")
        print(tens_power[l - 3], end=" ")

    l -= 1

else:

    if (ord(num[x]) - 48 == 1):
        sum = (ord(num[x]) - 48 +
              ord(num[x+1]) - 48)
        print(two_digits[sum])
        return

    elif (ord(num[x]) - 48 == 2 and
          ord(num[x + 1]) - 48 == 0):
        print("twenty")
        return

else:
    i = ord(num[x]) - 48
    if(i > 0):
        print(tens_multiple[i], end=" ")
    else:
        print("", end="")
    x += 1
    if(ord(num[x]) - 48 != 0):
        print(single_digits[ord(num[x]) - 48])
x += 1

```

```

# Driver Code
convert_to_words("1121")

```

## Q44: Counting Distinct Elements in an Array

Write a program for counting Distinct Elements in an Array

Code in C

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

```

int countDistinct(int *array, int size){
    int count = 0;

    // Increase count only at last occurrence of item in array
    for (int i = 0; i < size; i++){

        int j = 0;
        // traverse rightwards
        for (j = i+1; j < size; j++)
        {
            // if found duplicated found rightwards in the array
            if (array[i] == array[j]){
                break;
            }
        }
        // if traversed till the end of the array no break happenned
        // thus, this must have been last occurrence of arr[i]
        if (j == size)
            count++;
    }
    return count;
}

int main()
{
    int arr[] = {5, 8, 5, 7, 8, 10};
    int size = sizeof(arr)/sizeof(arr[0]);

    printf("Distinct items: %d",countDistinct(arr, size));

    return 0;
}

```

## Code in C++

```

#include <bits/stdc++.h>
using namespace std;

// Main function to run the program
int main()
{
    int arr[] = {10, 30, 10, 20, 40, 20, 50, 10};
    int n = sizeof(arr)/sizeof(arr[0]);

    int visited[n], count_dis=0;

```

```

        for(int i=0; i<n; i++){

            if(visited[i]!=1){
                for(int j=i+1; j<n; j++){
                    if(arr[i]==arr[j]){
                        visited[j]=1;
                    }
                }
                count_dis++;
            }
        }
        cout<<count_dis;
        return 0;
    }
}

```

## Code in Java

```

import java.util.Arrays;

class Main
{
    public static void countFreq(int arr[], int n)
    {
        boolean visited[] = new boolean[n];
        Arrays.fill(visited, false);
        int count_dis=0;
        // Traverse through array elements and
        // count frequencies
        for (int i = 0; i < n; i++) {

            // Skip this element if already processed
            if (visited[i] == true)
                continue;

            for (int j = i + 1; j < n; j++) {
                if (arr[i] == arr[j]) {
                    visited[j] = true;
                }
            }
            count_dis = count_dis+1;
        }
        System.out.println(count_dis);
    }
}

```

```

}

// Driver code
public static void main(String []args)
{
    int arr[] = new int[]{10, 30, 40, 20, 10, 20, 50, 10};
    int n = arr.length;
    countFreq(arr, n);
}
}

```

## Code in Python

```

def count(arr, n):

    # Mark all array elements as not visited
    visited = [False for i in range(n)]
    count_dis=0

    # Traverse through array elements
    # and count frequencies
    for i in range(n):
        # Skip this element if already
        # processed
        if (visited[i] == True):
            continue

        # Count frequency
        for j in range(i + 1, n, 1):
            if (arr[i] == arr[j]):
                visited[j] = True

        count_dis = count_dis+1;

    print(count_dis)

# Driver Code
arr = [10, 30, 40, 20, 10, 20, 50, 10]
n = len(arr)
count(arr, n)

```



## Q45 Counting number of days in a given month of a year

Write a program to count the number of days in a given month of a year.

### Code in C

```
#include <stdio.h>
int main()
{
    int month = 12, year=2012;

    if((month==2) && ((year%400==0) || ((year%100!=0)&&(year%4==0)))){
        printf("Number of days is 29");
    }
    else if(month==2){
        printf("Number of days is 28");
    }
    else if(month==1 || month==3 || month==5 || month==7 || month==8 || month==10 ||
month==12){
        printf("Number of days is 31");
    }
    else if(month==4 || month==6 || month==9 || month==11){
        printf("Number of days is 30");
    }
    else printf("Invalid month");

    return 0;
}
```

### Code in C++

```
#include <bits/stdc++.h>
using namespace std;

int main()
{
    int month = 12, year=2012;

    if((month==2) && ((year%400==0) || ((year%100!=0)&&(year%4==0)))){
        cout<<"Number of days is 29";
    }
    else if(month==2){
        cout<<"Number of days is 28";
    }
}
```

```

    else if(month==1 || month==3 || month==5 || month==7 || month==8 || month==10 ||
month==12){
        cout<<"Number of days is 31";
    }
    else if(month==4 || month==6 || month==9 || month==11){
        cout<<"Number of days is 30";
    }
    else cout<<"Invalid month";

    return 0;
}

```

## Code in Java

```

import java.io.*;
import java.util.*;

class Main{

    public static void main(String args[]){

        int month, year;
        month = 12;
        year = 2012;
        if((month==2) && ((year%4==0) || ((year%100!=0)&&(year%400==0))))
            System.out.println("Number of days is 29");

        else if(month==2)
            System.out.println("Number of days is 28");

        else if(month==1 || month==3 || month==5 || month==7 || month==8 || month==10 ||
month==12)
            System.out.println("Number of days is 31");

        else
            System.out.println("Number of days is 30");

    }

}

```

## Code in Python

```

month = 12

```

```
year=2012
```

```
if((month==2) and ((year%4==0) or ((year%100==0) and (year%400==0)))) :  
    print("Number of days is 29");
```

```
elif(month==2) :  
    print("Number of days is 28");
```

```
elif(month==1 or month==3 or month==5 or month==7 or month==8 or month==10 or  
month==12) :  
    print("Number of days is 31");
```

```
else :  
    print("Number of days is 30");
```

## Q46: Finding Repeating elements in an Array

Write a program for finding Repeating elements in an Array

### Code in C

```
#include<stdio.h>  
// Time Complexity :  $O(n^2)$   
//  $O(n^2)$  : sort the array and  $O(n)$  to count repeating  
  
// using bubble sort to sort the array  
// you can use a better sorting algorithm to reduce  
// time complexity to  $O(n \log n)$  to sort  
int bubbleSort(int arr[], int size){  
    for (int i = 0; i < size-1; i++){  
  
        // Since, after each iteration rightmost i elements are sorted  
        for (int j = 0; j < size-i-1; j++) if (arr[j] > arr[j+1])  
        {  
            int temp = arr[j]; // swap the element  
            arr[j] = arr[j+1];  
            arr[j+1] = temp;  
        }  
    }  
}  
  
// find repeating element
```

```

void findRepeating(int arr[], int n){
    // Traverse the sorted array
    // arr = {3, 10, 10, 11, 20, 20, 30, 30}

    int count = 0;
    // this step may look like O(n^2) but its actually O(n)
    // look closely, we traverse every element only once
    for (int i = 0; i < n; i++) {

        int flag = 0;

        // Move the index ahead whenever you encounter duplicates
        while (i < n - 1 && arr[i] == arr[i + 1]){
            // indicates that we have found duplicate
            flag = 1;
            i++;
        }
        // since i++ happened, we need to print previous element
        if(flag)
            printf("%d, ",(arr[i-1]));
    }

    return;
}

// Main function to run the program
int main()
{
    int arr[] = {20, 30, 10, 2, 10, 20, 30, 11};
    int n = sizeof(arr)/sizeof(arr[0]);

    bubbleSort(arr, n);

    findRepeating(arr, n);

    return 0;
}

```

## Code in C++

```

#include <bits/stdc++.h>
using namespace std;

```

```

// Main function to run the program
int main()
{
    int arr[] = {10, 30, 10, 20, 40, 20, 50, 10};
    int n = sizeof(arr)/sizeof(arr[0]);

    int visited[n];

    for(int i=0; i<n; i++){

        if(visited[i]!=1){
            int count = 1;
            for(int j=i+1; j<n; j++){
                if(arr[i]==arr[j]){
                    count++;
                    visited[j]=1;
                }
            }
            if(count!=1)
                cout<<arr[i]<<" ";
        }
    }

    return 0;
}

```

## Code in Java

```

import java.util.Arrays;

class Main
{
    public static void countFreq(int arr[], int n)
    {
        boolean visited[] = new boolean[n];
        Arrays.fill(visited, false);

        // Traverse through array elements and
        // count frequencies
        for (int i = 0; i < n; i++) {

            // Skip this element if already processed
            if (visited[i] == true)
                continue;

```

```

        // Count frequency
        int count = 1;
        for (int j = i + 1; j < n; j++) {
            if (arr[i] == arr[j]) {
                visited[j] = true;
                count++;
            }
        }
        if(count!=1)
            System.out.println(arr[i]);
    }
} // Driver code

public static void main(String []args)
{
    int arr[] = new int[]{10, 30, 30, 20, 10, 20, 50, 10};
    int n = arr.length;
    countFreq(arr, n);
}
}

```

## Code in Python

```

def count(arr, n):

    # Mark all array elements as not visited
    visited = [False for i in range(n)]

    # Traverse through array elements
    # and count frequencies
    for i in range(n):

        # Skip this element if already
        # processed
        if (visited[i] == True):
            continue

        # Count frequency
        count = 1
        for j in range(i + 1, n, 1):
            if (arr[i] == arr[j]):
                visited[j] = True
                count += 1
        if count != 1 :
            print(arr[i]);

```

```
# Driver Code
arr = [10, 30, 40, 20, 10, 20, 50, 10]
n = len(arr)
count(arr, n)
```

## Q47: Finding number of integers which has exactly x divisors

Write a program to find number of integers which has exactly x divisors.

### Code in C

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

int main(){

    int n=7, x=2;

    //Variable of count required numbers
    int count = 0;

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

        //variable to count the factors of i-th number
        int count_factors = 0;
        for(int j = 1; j<= i; j++){
            if(i%j==0){
                count_factors++;
            }
        }

        if(count_factors == x)
            count++;
    }

    printf("%d ", count);
}
```

### Code in C++

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

```

using namespace std;

int main(){

    int n=7, x=2;

    //Variable of count required numbers
    int count = 0;

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

        //variable to count the factors of i-th number
        int count_factors = 0;
        for(int j = 1; j<=sqrt(i); j++){
            if(i%j==0){
                if(i/j != j)
                    count_factors += 2;
                else
                    count_factors++;
            }
        }

        if(count_factors == x)
            count++;
    }

    cout<<count;
}

```

## Code in Java

```

import java.util.*;

class Main{
    public static void main(String[] args)
    {
        int n = 7, x = 2 ;
        int count = 0;

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

            //variable to count the factors of i-th number
            int count_factors = 0;
            for(int j = 1; j<= i; j++){
                if(i%j==0){

```



```

        count_factors++;
    }
}

if(count_factors == x)
    count++;
}
System.out.println(count);
}
}

```

## Code in Python

```

Number = 7
Divisor = 2
#count is to count total number of Numbers with exact divisor
count = 0
#driver loop
for i in range(1,Number+1):
    #count_factors checks the total number of divisors
    count_factors = 0
    #loop to find number of divisors
    for j in range(1,i+1):
        if i % j == 0:
            count_factors += 1
        else:
            pass
    if count_factors == Divisor:
        count +=1

#for break in line between Numbers and total count
print(count)

```

## Q48: Finding Roots of a quadratic equation

Write a program to find roots of a quadratic equation.

### Code in C

```

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

void findRoots(int a, int b, int c)

```

```

{
    if (a == 0) {
        printf("Invalid");
        return;
    }

    int d = b * b - 4 * a * c;
    double sqrt_val = sqrt(abs(d));

    if (d > 0) {
        printf("Roots are real and different \n");
        printf("%f\n%f", (double)(-b + sqrt_val) / (2 * a), (double)(-b - sqrt_val) / (2 * a));
    }
    else if (d == 0) {
        printf("Roots are real and same \n");
        printf("%f", -(double)b / (2 * a));
    }
    else // d < 0
    {
        printf("Roots are complex \n");
        printf("%f + i%f\n%f - i%f", -(double)b / (2 * a), sqrt_val/(2 * a), -(double)b / (2 * a),
sqrt_val/(2 * a));
    }
}

int main()
{
    int a = 1, b = 4, c = 4;

    findRoots(a, b, c);
    return 0;
}

```

## Code in C++

```

#include <bits/stdc++.h>
using namespace std;

```

```

void findRoots(int a, int b, int c)
{
    if (a == 0) {
        cout << "Invalid"; return; } int d = b * b - 4 * a * c; double sqrt_val = sqrt(abs(d)); if (d > 0) {
        cout << "Roots are real and different \n";
        cout << (double)(-b + sqrt_val) / (2 * a) << "\n" << (double)(-b - sqrt_val) / (2 * a);
    }
}

```

```

else if (d == 0) {
    cout << "Roots are real and same \n";
    cout << -(double)b / (2 * a);
}
else // d < 0
{
    cout << "Roots are complex \n";
    cout << -(double)b / (2 * a) << " + i" << sqrt_val<< "\n" << -(double)b / (2 * a) << " - i" <<
sqrt_val;
}
}

```

```

// Driver code
int main()
{
    int a = 1, b = 4, c = 4;

    findRoots(a, b, c);
    return 0;
}

```

## Code in Java

```

import java.io.*;
import static java.lang.Math.*;
class Main{

    static void findRoots(int a, int b, int c)
    {
        if (a == 0) {
            System.out.println("Invalid");
            return;
        }

        int d = b * b - 4 * a * c;
        double sqrt_val = sqrt(abs(d));

        if (d > 0) {
            System.out.println("Roots are real and different");
            System.out.println((double)(-b + sqrt_val) / (2 * a) + "\n" + (double)(-b - sqrt_val) / (2 * a));
        }
        else if (d == 0) {
            System.out.println("Roots are real and same ");
            System.out.println(-(double)b / (2 * a) + "\n" + -(double)b / (2 * a));
        }
    }
}

```

```

        else // d < 0
        {
            System.out.println("Roots are complex");

            System.out.println(-(double)b / (2 * a) + " + i" + sqrt_val + "\n" + -(double)b / (2 * a) + " - i"
+ sqrt_val);
        }
    }

    // Driver code
    public static void main(String args[])
    {

        int a = 1, b = 4, c = 4;

        // Function call
        findRoots(a, b, c);
    }
}

```

## Code in Python

```

import math

def findRoots(a, b, c):

    if a == 0:
        print("Invalid")
        return -1

    d = b * b - 4 * a * c
    sqrt_val = math.sqrt(abs(d))

    if d > 0:
        print("Roots are real and different ")
        print((-b + sqrt_val)/(2 * a))
        print((-b - sqrt_val)/(2 * a))
    elif d == 0:
        print("Roots are real and same")
        print(-b / (2*a))
    else: # d<0
        print("Roots are complex")
        print(- b / (2*a), " + i", sqrt_val)
        print(- b / (2*a), " - i", sqrt_val)

```

```
# Driver Program
```

```
a = 1
```

```
b = 4
```

```
c = 4
```

```
# Function call
```

```
findRoots(a, b, c)
```

## Q49: Removing Duplicate elements from an array

Write a program for removing Duplicate elements from an array

### Code in C

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

```
int duplicates(int arr[], int n)
```

```
{
```

```
    if (n==0 || n==1)
```

```
        return n;
```

```
    int temp[n];
```

```
    int j = 0;
```

```
    for (int i=0; i<n-1; i++)
```

```
        if (arr[i] != arr[i+1])
```

```
            temp[j++] = arr[i];
```

```
    temp[j++] = arr[n-1];
```

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

```
        arr[i] = temp[i];
```

```
    return j;
```

```
}
```

```
// Driver code
```

```
int main()
```

```
{
```

```
    int arr[] = {10, 20, 20, 30, 40, 40, 40, 50, 50};
```

```
    int n = sizeof(arr) / sizeof(arr[0]);
```

```
    n = duplicates(arr, n);
```

```

    for (int i=0; i<n; i++)
        printf("%d ", arr[i]);

    return 0;
}

```

## Code in C++

```

#include<bits/stdc++.h>

using namespace std;
int main(){
    set<int>s;
    int arr[] = {10, 10, 20, 30, 30, 30, 40};
    int n = sizeof(arr)/sizeof(arr[0]);
    for(int i=0; i<n; i++)
        s.insert(arr[i]);
    for(auto it = s.begin(); it != s.end(); it++)
        cout<<*it<<" ";
}

```

## Code in Java

```

import java.util.*;
class Main
{
    public static void main (String[] args)
    {
        int arr[] = {10, 20, 20, 30, 40, 40, 40, 50, 50};
        int n = arr.length;
        Set hash_Set = new HashSet();
        for (int i=0; i<n; i++)
            hash_Set.add(arr[i]);

        System.out.print(hash_Set);
    }
}

```

## Code in Python

```

def removeDuplicates(arr, n):
    # Return, if array is empty or contains a single element

```

```

if n == 0 or n == 1:
    return n

temp = list(range(n))

# Start traversing elements
j = 0;
for i in range(0, n-1):
    # If current element is not equal to next element then store that current element
    if arr[i] != arr[i+1]:
        temp[j] = arr[i]
        j += 1

temp[j] = arr[n-1]
j += 1

# Modify original array
for i in range(0, j):
    arr[i] = temp[i]

return j

# Driver code
arr = [10, 20, 20, 30, 40, 40, 40, 50, 50]
n = len(arr)

n = removeDuplicates(arr, n)

# Print updated array
for i in range(n):
    print ("%d"%(arr[i]), end = " ")

```

## Q50: Check whether a character is a vowel or consonant

Write a program to check whether a character is a vowel or consonant

### Code in C

```

#include <stdio.h>

// main function
int main()
{

```

```

char c='F';

if(c=='a' || c=='e' || c=='i' || c=='o' || c=='u' ||
c=='A' || c=='E' || c=='I' || c=='O' || c=='U')
{
    printf("%c is a vowel", c); // condition true input is vowel
}
else
{
    printf("%c is a consonant", c); // condition true input is consonant
}

return 0;
}

```

## Code in C++

```

#include<iostream>
using namespace std;

// main function
int main()
{
    char c;

    c='U';

    //checking for vowels
    if(c=='a' || c=='e' || c=='i' || c=='o' || c=='u' ||
c=='A' || c=='E' || c=='I' || c=='O' || c=='U')
    {
        cout << c << " is a vowel"; //condition true input is vowel
    }
    else
    {
        cout << c << " is a consonant"; //condition false input is consonant
    }

    return 0;
}

```

## Code in Java

```

public class Main
{

```



```

public static void main(String[] args)
{

    char c = 'g';

    //checking for vowels
    if(c=='a' || c=='e' || c=='i' || c=='o' || c=='u' || c=='A' || c=='E' || c=='I' || c=='O' || c=='U')
    {
        System.out.println(c + " is a vowel "); // condition true input is vowel
    }
    else
    {
        System.out.println(c + " is a consonant "); // condition true input is consonant
    }

}

}

```

## Code in Python

```

c = 'a'

# checking for vowels
if c == 'a' or c == 'e' or c == 'i' or c == 'o' or c == 'u' or c == 'A' or c == 'E' or c == 'I' or c == 'O' or c == 'U':
    print(c, "is a vowel") # condition true input is vowel
else:
    print(c, "is a consonant") # condition true input is consonant

```

## Q51: Check whether a character is a alphabet or not

Write a program to check whether a character is a alphabet or not

## Code in C

```

#include <stdio.h>
int main()
{
    char ch='9';

    if( (ch>='a' && ch<='z') || (ch>='A' && ch<='Z'))
        printf("The inserted character %c is an Alphabet", ch);
}

```

```

else
    printf("The entered character %c is not an Alphabet", ch);

return 0;
}

```

## Code in C++

```

#include <iostream>
using namespace std;
int main()
{
    char ch='9';

    if( (ch>='a' && ch<='z') || (ch>='A' && ch<='Z'))
        cout<<"The inserted character " <<ch<<" is an Alphabet";

    else
        cout<<"The inserted character " <<ch<<" is not an Alphabet";

    return 0;
}

```

## Code in Java

```

import java.util.Scanner;

//class declaration
public class Main
{

    //main method declaration
    public static void main(String[] args)
    {
        char ch;

        ch = '9';

        //condition for checking characters
        if((ch >= 'a' && ch <= 'z') || (ch >= 'A' && ch <= 'Z'))
            System.out.println("The inserted character " + ch + " is an Alphabet");
        else
            System.out.println("The inserted character " + ch + " is not an Alphabet");
    }
}

```

```
    } //end of the main method  
  
}
```

## Code in Python

```
ch = 'z'  
  
# basic logic  
if 'a' <= ch <= 'z' or 'A' <= ch <= 'Z':  
    print("The character", ch, "is an Alphabet")  
else:  
    print("The character", ch, "is not an Alphabet")
```

## Q52: Counting the number of even and odd elements in an array

Write a program for counting the number of even and odd elements in an array

## Code in C

```
#include<stdio.h>  
  
int main(){  
  
    int arr[] = {1, 7, 8, 4, 5, 16, 8};  
    int n = sizeof(arr)/sizeof(arr[0]);  
  
    int even_count=0, odd_count=0;  
  
    for(int i=0; i<n; i++){  
        if(arr[i]%2==0)  
            even_count++;  
  
        else  
            odd_count++;  
    }  
    printf("Even Elements count : %d \nOdd Elements count : %d", even_count, odd_count);  
}
```

## Code in C++

```
#include<bits/stdc++.h>
using namespace std;

int main(){

    int arr[] = {1, 7, 8, 4, 5, 16, 8};
    int n = sizeof(arr)/sizeof(arr[0]);

    int even_count=0, odd_count=0;

    for(int i=0; i<n; i++){
        if(arr[i]%2==0)
            even_count++;

        else
            odd_count++;
    }
    cout<<"Even Elements count : " <<even_count<< "\nOdd Elements count : " << odd_count;
}
```

## Code in Java

```
class Main{
    public static void main (String[] args)
    {
        int arr[] = {1, 20, 60, 31, 75, 40, 80};
        int n = arr.length;
        int countEven = 0, countOdd = 0;

        for(int i=0; i<n; i++){
            if((arr[i] % 2 )== 0)
                countEven += 1;

            else
                countOdd += 1;
        }
        System.out.println("Even Elements count : "+ countEven);
        System.out.println("Odd Elements count : "+ countOdd);
    }
}
```

## Code in Python

```
arr = [1, 7, 8, 4, 5, 16, 8]
n = len(arr)
countEven = 0
countodd = 0
for i in range(0, n):
    if arr[i]%2==0 :
        countEven += 1
    else:
        countodd += 1

print("Even Elements count : " )
print(countEven)

print("Odd Elements count : ")
print(countodd)
```

## Q53: Find the ASCII value of a character

Write a program to find the ASCII value of a character

### Code in C

```
#include <stdio.h>

int main()
{
    char a='5';
    printf("The ASCII value of %c is %d",a,a);

    return 0;
}
```

### Code in C++

```
#include <iostream>
using namespace std;

//main program
int main()
{
    char val='z';
```

```

        cout<<"The ASCII value of "<<val<<" is "<<(int)val;
        return 0;
    }

```

## Code in Java

```

import java.util.Scanner;
class Main
{
    public static void main(String[] args)
    {
        //scanner class object creation

        char c='A';

        //typecasting from character type to integer type
        int i = c;

        //printing ASCII value of the character
        System.out.println("ASCII value of "+c+" is "+i);

    }
}

```

## Code in Python

```

Char = 'z'

# convert Char to Ascii value
Ascii_val = ord(Char)

# print Value
print('The ASCII value of', Char, 'is', Ascii_val)

```

## Q54: Find all Symmetric pairs in an array

**Write program to find all Symmetric pairs in an array**

## Code in C

```

#include<stdio.h>

int main()

```

```

{
    int arr[5][2];
    arr[0][0] = 1; arr[0][1] = 2;
    arr[1][0] = 3; arr[1][1] = 4;
    arr[2][0] = 5; arr[2][1] = 1;
    arr[3][0] = 4; arr[3][1] = 3;
    arr[4][0] = 1; arr[4][1] = 5;

    for(int i=0; i<5; i++){
        for(int j=i+1; j<5; j++){
            if(arr[i][0]==arr[j][1] && arr[i][1]==arr[j][0])
                printf("(%d, %d) ", arr[i][0], arr[i][1]);
        }
    }

    return 0;
}

```

## Code in C++

```

#include<bits/stdc++.h>
using namespace std;

int main ()
{
    int arr[5][2];
    arr[0][0] = 1;
    arr[0][1] = 2;
    arr[1][0] = 3;
    arr[1][1] = 4;
    arr[2][0] = 5;
    arr[2][1] = 1;
    arr[3][0] = 4;
    arr[3][1] = 3;
    arr[4][0] = 1;
    arr[4][1] = 5;

    for (int i = 0; i < 5; i++){
        for (int j = i + 1; j < 5; j++){
            if (arr[i][0] == arr[j][1] && arr[i][1] == arr[j][0])
                cout << "(" << arr[i][0] << ", " << arr[i][1] << ") ";
        }
    }

    return 0;
}

```

```
}
```

## Code in Java

```
class Main{

    public static void main(String arg[])
    {
        int arr[][] = new int[5][2];

        arr[0][0] = 1;
        arr[0][1] = 2;
        arr[1][0] = 3;
        arr[1][1] = 4;
        arr[2][0] = 5;
        arr[2][1] = 1;
        arr[3][0] = 4;
        arr[3][1] = 3;
        arr[4][0] = 1;
        arr[4][1] = 5;

        for (int i = 0; i < 5; i++){
            for (int j = i + 1; j < 5; j++){
                if (arr[i][0] == arr[j][1] && arr[i][1] == arr[j][0])
                    System.out.println("(" + arr[i][0] + ", " + arr[i][1] + ")");
            }
        }

    }
}
```

## Code in Python

```
# Function to find all pairs that are a mirror of each other
def findPairs(pairs):
```

```
    # create an empty set of strings
    s = set()
```

```
    # do for each pair
    for (x, y) in pairs:
```

```
        # insert the current pair `(x, y)` into the set
        s.add((x, y))
```



```
# if mirror pair `(y, x)` is seen before, print the pairs
if (y, x) in s:
    print((x, y))
```

```
pairs = [(3, 4), (1, 2), (5, 2), (7, 10), (4, 3), (2, 5)]
findPairs(pairs)
```

## Q55: Length of the string without using strlen() function

Write a program to find the length of the string without using strlen() function

### Code in C

```
#include <stdio.h>
int main()
{
    //Initializing variable.
    char str[100];
    int i,length=0;

    //Accepting input.
    printf("Enter a string: \n");
    scanf("%s",str);

    //Initializing for loop.
    for(i=0; str[i]!='\0'; i++)
    {
        length++; //Counting the length.
    }

    printf("\nLength of input string: %d",length);

    return 0;
}
```

### Code in C++

```
#include <iostream>
using namespace std;

int main()
```

```

{
    //Initializing variable.
    char str[30];
    int i,length=0;
    //Accepting input.
    cout<<"Enter the string:";
    cin>>str;
    //Initializing for loop.
    for(i=0;str[i]!='\0';++i)
    {
        length++;//Counting the length.
    }

    cout<<"Length of the string is:"<<length<<endl;

    return 0;
}

```

## Code in Java

```

import java.util.Scanner;

public class Main {

    public static void main(String[] args) {
        int length=0;

        String s = "prepinsta";
        for (char c1 : s.toCharArray())
            length++;
        System.out.println("Length of String is : "+length);

    }

}

```

## Code in Python

```

string = 'Hello'

count = 0

for i in string:

    count+=1

```

```
print(count)
```

## Q56: Determine Array is a subset of another array or not

Write a program to determine Array is a subset of another array or not

Code in C

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

```
int isSubset(int arr1[], int arr2[], int m, int n)
```

```
{
    int i = 0;
    int j = 0;
    for (i = 0; i < n; i++) {
        for (j = 0; j < m; j++) {
            if (arr2[i] == arr1[j])
                break;
        }

        if (j == m)
            return 0;
    }
}
```

```
    return 1;
```

```
}
```

```
// Driver code
```

```
int main()
```

```
{
    int arr1[] = { 11, 10, 13, 21, 30, 70 };
    int arr2[] = { 11, 30, 70, 10 };
```

```
    int m = sizeof(arr1) / sizeof(arr1[0]);
    int n = sizeof(arr2) / sizeof(arr2[0]);
```

```
    if (isSubset(arr1, arr2, m, n))
        printf("arr2[] is subset of arr1[] ");
    else
        printf("arr2[] is not a subset of arr1[]");
```

```
    return 0;
```

```
}
```

## Code in C++

```
#include <bits/stdc++.h>
using namespace std;

bool isSubset(int arr1[], int arr2[], int m, int n)
{
    int i = 0;
    int j = 0;
    for (i = 0; i < n; i++) {
        for (j = 0; j < m; j++) {
            if (arr2[i] == arr1[j])
                break;
        }

        if (j == m)
            return 0;
    }

    return 1;
}

// Driver code
int main()
{
    int arr1[] = { 11, 10, 13, 21, 30, 70 };
    int arr2[] = { 11, 30, 70, 10 };

    int m = sizeof(arr1) / sizeof(arr1[0]);
    int n = sizeof(arr2) / sizeof(arr2[0]);

    if (isSubset(arr1, arr2, m, n))
        cout<<"arr2[] is subset of arr1[] ";
    else
        cout<<"arr2[] is not a subset of arr1[]";

    return 0;
}
```

## Code in Java

```

class Main{

    static boolean isSubset(int arr1[], int arr2[], int m, int n)
    {
        int i = 0;
        int j = 0;
        for (i = 0; i < n; i++) {
            for (j = 0; j < m; j++) {
                if (arr2[i] == arr1[j])
                    break;
            }

            if (j == m)
                return false;
        }

        return true;
    }

    public static void main(String args[])
    {
        int arr1[] = { 11, 10, 13, 21, 30, 70 };
        int arr2[] = { 11, 30, 70, 10 };

        int m = arr1.length;
        int n = arr2.length;

        if (isSubset(arr1, arr2, m, n))
            System.out.print("arr2[] is subset of arr1[] ");
        else
            System.out.print("arr2[] is not subset of arr1[] ");

    }
}

```

## Code in Python

```

def isSubset(arr1, arr2, m, n):
    i = 0
    j = 0
    for i in range(n):
        for j in range(m):

```

```

        if(arr2[i] == arr1[j]):
            break

    if (j == m):
        return 0
    return 1

# Driver code
arr1 = [11, 12, 13, 21, 30, 70]
arr2 = [11, 30, 70, 12]

m = len(arr1)
n = len(arr2)

if(isSubset(arr1, arr2, m, n)):
    print("arr2[] is subset of arr1[] ")
else:
    print("arr2[] is not a subset of arr1[]")

```

## Q57: Count the number of vowels

Write a program to count the number of vowels

### Code in C

```

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

int main()
{
    // Initializing variable.
    char str[100];
    int i, vowels = 0;

    // Accepting input.
    printf("Enter the string: ");
    // best way to read string rather than gets/fgets
    scanf("%[^\n]s",&str);

    //Initializing for loop.
    for(i = 0; str[i]; i++)
    {
        //Counting the vowels.

```

```

        if(str[i]=='a' || str[i]=='e' || str[i]=='i' ||
           str[i]=='o' || str[i]=='u' || str[i]=='A' ||
           str[i]=='E' || str[i]=='I' || str[i]=='O' || str[i]=='U')
        {
            vowels++;
        }
    }

    //Printing the count of vowels.
    printf("Total number of vowels: = %d\n",vowels);

    return 0;
}

```

## Code in C++

```

#include <iostream>
#include <stdio.h>

using namespace std;

int main()
{
    char str[100] = "prepinsta";
    int vowels = 0;

    // can also do str[i] != '\0' in condition below both would work
    for(int i = 0; str[i]; i++)
    {
        if(str[i]=='a' || str[i]=='e' || str[i]=='i' || str[i]=='o' || str[i]=='u'
           || str[i]=='A' || str[i]=='E' || str[i]=='I' || str[i]=='O' || str[i]=='U')
        {
            vowels++;
        }
    }

    cout << "Total Vowels : " << vowels;

    return 0;
}

```

## Code in Java

```

import java.util.Scanner;

```

```

public class Main{

    public static void main(String[] args) {
        String s = "prepinsta";
        char[] c = s.toCharArray();
        int vowel=0;

        for (int i = 0; i < s.length(); i++) {
            if(s.charAt(i)=='a' || s.charAt(i)=='e' || s.charAt(i)=='i' || s.charAt(i)=='o' || s.charAt(i)=='u')
                vowel++;
        }

        System.out.println("Vowels: " + vowel);
    }
}

```

## Code in Python

```

String = input('Enter the string :')
count = 0
#to check for less conditions
#keep string in lowercase
String = String.lower()
for i in String:
    if i == 'a' or i == 'e' or i == 'i' or i == 'o' or i == 'u':
        #if True
        count+=1
#check if any vowel found
if count == 0:
    print('No vowels found')
else:
    print('Total vowels are :' + str(count))

```

## Q58: Remove the vowels from a String

Write a program to remove the vowels from a String

## Code in C

```

#include <stdio.h>
#include <stdlib.h>

int main() {
    // Initializing variable.

```



```

char str[100];
int i, j, len = 0;

// Accepting input.
printf("Enter a string : ");
// gets(str);
scanf("%s", str);

len = strlen(str);
// Accepting input.

for (i = 0; i < len; i++) {
    // Checking vowels.
    if (str[i] == 'a' || str[i] == 'e' || str[i] == 'i' || str[i] == 'o' || str[i] == 'u' ||
        str[i] == 'A' || str[i] == 'E' || str[i] == 'I' || str[i] == 'O' || str[i] == 'U') {
        // Deleting vowels.
        for (j = i; j < len; j++) {
            // Storing string without vowels.
            str[j] = str[j + 1];
        }
        i--;
        len--;
    }
    str[len + 1] = '\0';
}
printf("After deleting the vowels, the string will be : %s", str);
return 0;
}

```

## Code in C++

```

#include <bits/stdc++.h>

using namespace std;

int main()
{
    char str[100];

    cout << "Enter a string : "; cin >> str;

    int len = strlen(str);

    for(int i=0; i<len; i++)
    {

```

```

        // checking vowels.
        if(str[i]=='a' || str[i]=='e' || str[i]=='i' || str[i]=='o' || str[i]=='u'
           || str[i]=='A' || str[i]=='E' || str[i]=='I' || str[i]=='O' || str[i]=='U')
        {

                for(int j=i; j<len; j++)
                {
                        str[j]=str[j+1];
                }

                i--;
                len--;
        }
    }
    cout << "After removing Vowels: " << str;

    return 0;
}

```

## Code in Java

```

import java.util.Scanner;

public class Main {

    public static void main(String[] args) {

        String s = "prepinsta";
        String s1 = "";
        s1 = s.replaceAll("[aeiou]", "");
        System.out.println("String after removing vowel : "+s1);
    }

}

```

## Code in Python

```

string = "Preplnsta"

vowels = ['a', 'e', 'i', 'o', 'u', 'A', 'E', 'I', 'O', 'U']
result = ""

for i in range(len(string)):
    if string[i] not in vowels:
        result = result + string[i]

```

```
print("\nAfter removing Vowels: ", result)
```

## Q59: Replace each element of the array by its rank in the array

Write a program to Replace each element of the array by its rank in the array

### Code in C

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

```
int main(){
    int arr[] = { 100, 2, 70, 12 , 90};
    int n = sizeof(arr) / sizeof(arr[0]);
```

```
    int temp[n];
    for(int i=0; i<n; i++)
        temp[i] = arr[i];
```

```
    //sort the copied array
    for(int i=0; i<n; i++){
        for(int j=i+1; j<n; j++){
            int x = temp[i];
            temp[i] = temp[j];
            temp[j] = x;
        }
    }
```

```
    for(int i=0; i<n; i++){

        for(int j=0; j<n; j++){
            if(temp[j]==arr[i])
            {
                arr[i] = j+1;
                break;
            }
        }
    }
```

```
    for(int i=0; i<n; i++)
        printf("%d ", arr[i]);
}
```

## Code in C++

```
#include<bits/stdc++.h>
using namespace std;

int main(){
    int arr[] = { 100, 2, 70, 12 , 90};
    int n = sizeof(arr) / sizeof(arr[0]);

    int temp[n];
    for(int i=0; i<n; i++)
        temp[i] = arr[i];

    //sort the copied array
    sort(temp, temp+n);

    for(int i=0; i<n; i++){

        for(int j=0; j<n; j++){
            if(temp[j]==arr[i])
            {
                arr[i] = j+1;
                break;
            }
        }
    }

    for(int i=0; i<n; i++)
        cout<<arr[i]<<" ";
}
```

## Code in Java

```
import java.util.*;

class Main {

    static void changeArr(int[] input)
    {
        // Copy input array into newArray
        int newArray[] = Arrays.copyOfRange(input, 0, input.length);

        // Sort newArray[] in ascending order
        Arrays.sort(newArray);
        for(int i=0; i< input.length; i++){
```

```

        for(int j=0; j< input.length; j++){
            if(newArray[j]==input[i])
            {
                input[i] = j+1;
                break;
            }
        }
    }
}

// Driver Code
public static void main(String[] args)
{
    // Given array arr[]
    int[] arr = { 100, 2, 70, 12 , 90};

    // Function Call
    changeArr(arr);

    // Print the array elements
    System.out.println(Arrays.toString(arr));
}
}

```

## Code in Python

```

def changeArr(input1):

    newArray = input1.copy()
    newArray.sort()

    for i in range(len(input1)):
        for j in range(len(newArray)):
            if input1[i]==newArray[j]:
                input1[i] = j+1;
                break;

# Driver Code
arr = [100, 2, 70, 12 , 90]
changeArr(arr)
# Print the array elements
print(arr)

```

## Q60: Check if the given string is Palindrome or not

Write a program to check if the given string is Palindrome or not

### Code in C

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

int main()
{
    char str[10] = "naman";
    int i, len, flag = 0;

    len = strlen(str);

    for (i = 0; i < len; i++)
    {
        // Checking if string is palindrome or not
        if (str[i] != str[len - i - 1]) {
            flag = 1;
            break;
        }
    }

    if (flag)
        printf("%s is not palindrome", str);
    else
        printf("%s is palindrome", str);

    return 0;
}
```

### Code in C++

```
#include <bits/stdc++.h>
using namespace std;
void Lower_case(char str[]) {
    int i = 0;
    while (str[i] != '\0')
    {
        if (str[i] > 64 && str[i] < 91)
            str[i] += 32;
        i++;
    }
}
```

```

    }
}
// A function to check if a string str is palindrome
void CheckPalindrome(char str[]) {
// Start from leftmost and rightmost corners of str
int l = 0;
int h = strlen(str) - 1;

// Keep comparing characters while they are same
Lower_case(str);

while (h > l) {
    if (str[l++] != str[h--]) {
        cout << "The String " << str << " is not a palindrome\n";
        return;
    }
}
cout << "The String " << str << " is a palindrome";
}
// Driver program to test CheckPalindrome function
int main() {
char str1[50] = "naman"; // size of char string
CheckPalindrome(str1);
cout<<endl;
char str2[50] = "radar"; // size of char string
CheckPalindrome(str2);
return 0;
}

```

## Code in Java

```

import java.util.Scanner;

public class Main {

    public static void main(String[] args) {

        String s = "arora";
        String rev = "";
        for (int i = s.length()-1; i >=0 ; i--)
            rev=rev+s.charAt(i);
        if(s.equals(rev))
            System.out.println("String is palindrome");
        else
            System.out.println("String is not palindrome");
    }
}

```

```
    }  
}
```

### Code in Python

```
input_string = 'civic'  
rev = input_string[::-1]  
  
if input_string == rev:  
    print(rev + " is Palindrome")  
else:  
    print(rev + " is not Palindrome")
```

## Q61: Print the given string in reverse order

Write program to print the given string in reverse order

### Code in C

```
#include <stdio.h>  
  
#include <string.h>  
  
// function definition of the revstr()  
void revstr(char *str1)  
{  
    // declare variable  
    int i, len, temp;  
    len = strlen(str1); // use strlen() to get the length of str string  
  
    // use for loop to iterate the string  
    for (i = 0; i < len/2; i++)  
    {  
        // temp variable use to temporary hold the string  
        temp = str1[i];  
        str1[i] = str1[len - i - 1];  
        str1[len - i - 1] = temp;  
    }  
}  
  
int main()  
{
```



```

char str[50]="priyanka"; // size of char string

printf (" \n Before reversing the string: %s \n", str);

// call revstr() function
revstr(str);
printf (" After reversing the string: %s", str);
}

```

## Code in C++

```

#include <iostream>
#include <string.h>
using namespace std;

// function definition of the revstr()
void revstr(char *str1) {
    // declare variable
    int i, len, temp;
    len = strlen(str1);

    // use strlen() to get the length of str string
    // use or loop to iterate the string
    for(i = 0; i < len/2; i++)
    {
        //temp variable use to temporary hold the string
        temp = str1[i];
        str1[i] = str1[len - i - 1];
        str1[len - i - 1] = temp;
    }
}

int main()
{
    char str[50] = "Priyanka";
    cout << "Before reversing the string: " << str;

    revstr(str);

    cout<< "\nAfter reversing the string: " << str;
    return 0;
}

```

## Code in Java

```

import java.util.Scanner;

```

```

public class Main {
    public static void main(String[] args) {
        String s ="Prepinsta";
        StringBuilder sb = new StringBuilder();
        sb.append(s);
        System.out.println( "String is : "+sb);
        sb.reverse();
        System.out.println("Reversed string is : "+sb);
    }
}

```

## Code in Python

```

string = "Hello world"
print(string[::-1])

```

## Q62: Finding equilibrium index of an array

Write a program for finding equilibrium index of an array

### Code in C

```

#include<stdio.h>

void find (int arr[], int n)
{
    int result = -1;

    for(int i=0; i<n; i++){

        int left_sum =0;
        for(int j=0; j<i; j++)
            left_sum += arr[j];
        int right_sum =0;
        for(int j=i+1; j<n; j++)
            right_sum += arr[j];

        if(right_sum == left_sum)
            result = i;
    }

    printf("First Point of equilibrium is at index = %d\n", result);
    return;
}

```

```
}
```

```
int main ()  
{  
    int arr[]={4, -2, 0, 6, -4};  
    int n = sizeof(arr)/sizeof(arr[0]);  
  
    find (arr, n);  
    return 0;  
}
```

## Code in C++

```
#include<bits/stdc++.h>  
using namespace std;  
  
void find (int arr[], int n)  
{  
    int result = -1;  
  
    for(int i=0; i<n; i++){  
  
        int left_sum =0;  
        for(int j=0; j<i; j++)  
            left_sum += arr[j];  
        int right_sum =0;  
        for(int j=i+1; j<n; j++)  
            right_sum += arr[j];  
  
        if(right_sum == left_sum)  
            result = i;  
  
    }  
  
    cout<<"First Point of equilibrium is at index = "<<result;  
    return;  
  
}  
  
int main ()  
{  
    int arr[]={4, -2, 0, 6, -4};  
    int n = sizeof(arr)/sizeof(arr[0]);
```

```
    find (arr, n);  
    return 0;  
}
```

## Code in Java

```
import java.util.*;  
  
public class Main  
{  
    static int equilibrium_index(int arr[], int n)  
    {  
        int result = -1;  
  
        for(int i=0; i<n; i++){  
  
            int left_sum =0;  
            for(int j=0; j<i; j++)  
                left_sum += arr[j];  
  
            int right_sum =0;  
            for(int j=i+1; j<n; j++)  
                right_sum += arr[j];  
  
            if(right_sum == left_sum)  
                return i;  
  
        }  
  
        return -1;  
    }  
  
    public static void main(String[] args)  
    {  
        int arr[] = { 1,2,3,4,5,1,3,2,4 };  
        int arr_size = arr.length;  
        System.out.print("Equilibrium Index : ");  
        System.out.println(equilibrium_index(arr, arr_size));  
    }  
}
```

## Code in Python

```
def equilibrium(arr):
    leftsum = 0
    rightsum = 0
    n = len(arr)

    for i in range(n):
        leftsum = 0
        rightsum = 0

        for j in range(i):
            leftsum += arr[j]

        for j in range(i + 1, n):
            rightsum += arr[j]

        if leftsum == rightsum:
            return i

    return -1

# Driver code
arr = [-4, 1, 5, 2, -4, 4, 2]
print ('Equilibrium index is ',equilibrium(arr))
```

## Q63: Remove all characters from string except alphabets

Write a program to remove all characters from string except alphabets

### Code in C

```
#include <stdio.h>
int main()
{
    //Initializing variable.
    char str[100];
    int i, j;

    //Accepting input.
    printf(" Enter a string : ");
    gets(str);
```

```

//Iterating each character and removing non alphabetical characters.
for(i = 0; str[i] != '\0'; ++i)
{
    while (!( (str[i] >= 'a' && str[i] <= 'z') || (str[i] >= 'A' && str[i] <= 'Z') || str[i] == '\0') )
    {
        for(j = i; str[j] != '\0'; ++j)
        {
            str[j] = str[j+1];
        }
        str[j] = '\0';
    }
}
//Printing output.
printf(" After removing non alphabetical characters the string is :");
puts(str);
return 0;
}

```

## Code in C++

```

#include <iostream>
using namespace std;
int main()
{
    //Initializing variable.
    char str[100];
    int i, j;

    //Accepting input.
    cout<<"Enter a string : ";
    cin>>str;

    //Iterating each character and removing non alphabetical characters.
    for(i = 0; str[i] != '\0'; ++i)
    {
        while (!( (str[i] >= 'a' && str[i] <= 'z') || (str[i] >= 'A' && str[i] <= 'Z') || str[i] == '\0') )
        {
            for(j = i; str[j] != '\0'; ++j)
            {
                str[j] = str[j+1];
            }
            str[j] = '\0';
        }
    }
    //Printing output.
}

```

```

    cout<<"After removing non alphabetical characters the string is :";
    puts(str);
    return 0;
}

```

## Code in Java

```

class Main
{
    static void removeSpecialCharacter(String s)
    {
        for (int i = 0; i < s.length(); i++)
        {
            if (s.charAt(i) < 'A' || s.charAt(i) > 'Z' && s.charAt(i) < 'a' || s.charAt(i) > 'z')
            {
                s = s.substring(0, i) + s.substring(i + 1);
                i--;
            }
        }
        System.out.print(s);
    }
    // Driver code
    public static void main(String[] args)
    {
        String s = "$P*r;e..pi, ns'ta^?";
        removeSpecialCharacter(s);
    }
}

```

## Code in Python

```

String1 = "#Justice!For@Chutki123"
#initialize empty String
String2 = ""
for i in String1:
    #check for alphabets
    if (ord(i) >= 65 and ord(i) <= 90) or (ord(i) >= 97 and ord(i) <= 122):
        #concatenate to empty string
        String2+=i
print('Alphabets in string are :'+ String2)

```

## Q64: Find the “Kth” max and min element of an array

Write a program to Find the “Kth” max and min element of an array

### Code in C

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

void merge(int arr[], int l, int m, int r)
{
    int i, j, k;
    int n1 = m - l + 1;
    int n2 = r - m;

    int L[n1], R[n2];

    for (i = 0; i < n1; i++)
        L[i] = arr[l + i];
    for (j = 0; j < n2; j++)
        R[j] = arr[m + 1 + j];

    i = 0;
    j = 0;
    k = l;

    while (i < n1 && j < n2) {
        if (L[i] <= R[j]) {
            arr[k] = L[i];
            i++;
        }
        else {
            arr[k] = R[j];
            j++;
        }
        k++;
    }

    while (i < n1) {
        arr[k] = L[i];
        i++;
        k++;
    }

    while (j < n2) {
```



```

        arr[k] = R[j];
        j++;
        k++;
    }
}

void mergeSort(int arr[], int l, int r)
{
    if (l < r) {
        int m = l + (r - l) / 2;

        mergeSort(arr, l, m);
        mergeSort(arr, m + 1, r);

        merge(arr, l, m, r);
    }
}

int main()
{
    int arr[] = { 12, 11, 13, 5, 6, 7 };
    int arr_size = sizeof(arr) / sizeof(arr[0]);
    int k = 3;

    mergeSort(arr, 0, arr_size - 1);

    printf("K-th Maximum element : %d\n", arr[arr_size-k]);
    printf("K-th Minimum element : %d", arr[k-1]);

    return 0;
}

```

## Code in C++

```

#include <iostream>
#include <vector>
#include <set>
#include <iterator>

using namespace std;

int main() {
    int n;
    cin >> n;

```

```

vector<int> arr(n);

for (int i = 0; i < n; i++)
    cin >> arr[i];

int k;
cin >> k;

set<int> s(arr.begin(), arr.end());

set<int>::iterator itr = s.begin();

advance(itr, k - 1); // itr points to kth element (minimum) in set

cout << "Minimum: " << *itr << "\n";

itr = s.begin();

advance(itr, n - k); // itr points to kth element (maximum) in set

cout << "Maximum: " << *itr << "\n";

return 0;
}

```

## Code in Java

```

import java.io.*;
import java.util.*;

public class Main {
    public static void main(String[] args) throws Exception {
        Scanner sc = new Scanner(System.in);
        int n, k, i;

        System.out.println("Enter the size of the array: ");
        n = 3;

        System.out.println("Enter the elements for the array: ");
        int arr[] = {1,2,4,5};

        System.out.println("Enter the value of k: ");
    }
}

```

```

k = 2;

PriorityQueue queue = new PriorityQueue<>(Collections.reverseOrder());
System.out.println("Kth smallest element is: ");

for (i = 0; i < n; i++) {
    queue.add(arr[i]);
    if (queue.size() > k) {
        queue.poll(); // top elements will be removed if size>k
    }
}
System.out.println(queue.peek()); // top will be printed

PriorityQueue queue1 = new PriorityQueue<>();
System.out.println("Kth Largest element is: ");

for (i = 0; i < n; i++) {
    queue1.add(arr[i]);
    if (queue1.size() > k) {
        queue1.poll(); // top elements will be removed if size>k
    }
}
System.out.println(queue1.peek()); // top will be printed
}

```

## Code in Python

```

n = int(input("Enter the number of elements: "))
arr = [int(x) for x in input("Enter the elements separated by spaces: ").split()]
k = int(input("Enter the value of k: "))

# Using set to get unique elements and sorting the list
unique_elements = list(set(arr))
unique_elements.sort()

if k <= len(unique_elements):
    kth_min = unique_elements[k - 1]
    kth_max = unique_elements[-k]

    print(f"Minimum: {kth_min}")
    print(f"Maximum: {kth_max}")
else:
    print("k is out of range.")

```

## Q65: Remove spaces from a string

Write a program to remove spaces from a string

### Code in C

```
#include<stdio.h>
// Function to remove all spaces from a given string
void removeSpaces(char *str)
{
    // To keep track of non-space character count
    int count = 0;
    // Traverse the provided string. If the current character is not a space,
    //move it to index 'count++'.
    for (int i = 0; str[i]; i++)
        if (str[i] != ' ')
            str[count++] = str[i]; // here count is incremented
    str[count] = '\0';
}
// Driver program to test above function
int main()
{
    char str[] = "P re p i n sta ";
    removeSpaces(str);
    printf("%s", str);
    return 0;
}
```

### Code in C++

```
#include <iostream>
using namespace std;
// Function to remove all spaces from a given string
void removeSpaces(char *str)
{
    // To keep track of non-space character count
    int count = 0;
    // Traverse the provided string. If the current character is not a space,
    //move it to index 'count++'.
    for (int i = 0; str[i]; i++)
        if (str[i] != ' ')
            str[count++] = str[i]; // here count is
```

```

// incremented
str[count] = '\0';
}
// Driver program to test above function
int main()
{
char str[] = "P re p i n sta ";
removeSpaces(str);
cout << str;
return 0;
}

```

## Code in Java

```

import java.util.Scanner;

public class Main {

    public static void main(String[] args) {
        Scanner sc =new Scanner(System.in);
        String s = "Prepinsta is best";
        char[] c = s.toCharArray();
        StringBuffer sb = new StringBuffer();

        for (int i = 0; i < c.length; i++) {
            if( (c[i] != ' ') && (c[i] != '\t' )) {
                sb.append(c[i]);
            }
        }
        System.out.println("String after removing spaces : "+sb);
    }
}

```

## Code in Python

```

String = "PrepInsta is fabulous"

#Use join function
String = "".join(String.split())

print("After removing spaces string is :",String)

```

## Q66: Remove brackets from an algebraic expression

Write a program to remove brackets from an algebraic expression

### Code in C

```
#include<stdio.h>
int main()
{
    //Initializing variables.
    char str[100] = "Prep))insta", str_no_spc[100];
    int i=0, j=0 ;

    //Iterating each character of string.
    while(str[i] != '\0')
    {
        if(str[i] != '(' && str[i] != ')')//Excluding brackets.
        {
            str_no_spc[j++] = str[i];
        }
        i++;
    }
    str_no_spc[j] = '\0';

    //Printing result.
    printf("The string after removing all the spaces is:\n%s", str_no_spc);
    return 0;
}
```

### Code in C++

```
#include <iostream>
using namespace std;

int main()
{
    //Initializing variables.
    char str[100]="Prepins))ta", str_without_brackets[100];
    int i=0, j=0 ;

    //Iterating each character of string.
    while(str[i] != '\0')
    {
        if(str[i] != '(' && str[i] != ')')//Removing brackets.
        {

```

```

str_without_brackets[j++] = str[i];
}
i++;
}
str_without_brackets[j] = '\0';

//Printing result.
cout<<"The string after removing all the brackets is:\n"<<str_without_brackets;
return 0;
}

```

## Code in Java

```

import java.util.Scanner;

public class Main {

    public static void main(String[] args) {

        String s = "(a+b)=c";
        String result = s.replaceAll("[(){}]", "");
        System.out.println("Expression without brackets : "+result);
    }
}

```

## Code in Python

```

Exp = "(a-b)+[c*d]+{e/f}"
#initialize an empty string
Equation = ""
#traversing through string
for i in Exp:
    #checking for brackets
    if ord(i) == 41 or ord(i) == 40 or ord(i) == 91 or ord(i) == 93 or ord(i) == 123 or ord(i) == 125:
        #If True
        pass
    else:
        #if False
        #add it to empty String
        Equation = Equation + i
#print the string
print(' String without bracket is ' + Equation)

```

## Q67: Move all the negative elements to one side of the array

Write a program to move all the negative elements to one side of the array

### Code in C

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

```
void shiftall(int arr[], int left, int right)
{
```

```
    while (left<=right)
```

```
    {
```

```
        if (arr[left] < 0 && arr[right] < 0)
            left+=1;
```

```
        else if (arr[left]>0 && arr[right]<0)
```

```
        {
```

```
            int temp=arr[left];
            arr[left]=arr[right];
            arr[right]=temp;
            left+=1;
            right-=1;
```

```
        }
```

```
        else if (arr[left]>0 && arr[right] >0)
```

```
            right-=1;
```

```
    else{
```

```
        left += 1;
```

```
        right -= 1;
```

```
    }
```

```
    }
```

```
}
```

```
void display(int arr[], int right){
```

```
    for (int i=0;i<=right;++i){
```

```
        printf("%d ",arr[i]);
```

```
    }
```

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

```
}
```

```
int main()
```

```
{
```



```

int n;
scanf("%d", &n);

int arr[n];

for(int i=0; i<n; i++)
    scanf("%d", &arr[i]);

// Function Call
shiftall(arr,0,n-1);
display(arr,n-1);
return 0;
}

```

## Code in C++

```

#include<iostream>
using namespace std;

// Function to shift all the
// negative elements on left side
void shiftall (int arr[], int left, int right)
{
    while (left <= right)
    {
        if (arr[left] < 0 && arr[right] < 0)
            left += 1;
        else if (arr[left] > 0 && arr[right] < 0)
        {
            int temp = arr[left];
            arr[left] = arr[right];
            arr[right] = temp;
            left += 1; right -= 1;
        }
        else if (arr[left] > 0 && arr[right] > 0)
            right -= 1;

        else
        {
            left += 1;
            right -= 1;
        }
    }
}

```

```

}

void display (int arr[], int right)
{

    for (int i = 0; i <= right; ++i)
        {
            cout << arr[i] << " ";
        }
    cout << endl;
}
int main () {
    int n;
    cin >> n;

    int arr[n];

    for (int i = 0; i < n; i++) cin >> arr[i];

    // Function Call
    shiftall (arr, 0, n - 1);
    display (arr, n - 1);
    return 0;
}

```

## Code in Java

```

import java.util.*;
public class Main {
    public static void shift(int[] arr) {
        int j = 0;
        for (int i = 0; i < arr.length; i++) {
            if (arr[i] < 0) {
                if (i != j) swap(arr, i, j);
                j++;
            }
        }
        for (int i = 0; i < arr.length; i++) {
            System.out.print(arr[i] + " ");
        }
    }
    // used for swapping ith and jth elements of array
    public static void swap(int[] arr, int i, int j) {

```

```

        int temp = arr[i];
        arr[i] = arr[j];
        arr[j] = temp;
    }
    public static void main(String[] args) throws Exception {

        int[] arr = new int[]{ -1,0,3,1,-4};

        shift(arr);
    }
}

```

## Code in Python

```

def find(arr):
    # sort array
    arr.sort()

    # print array
    print("Array after moving all the elements to left:", arr)

array = [1, 3, -1, 4, -3, -5, -6, 3, 7]
# call function
find(array)

```

## Q68: Count the sum of numbers in a string

Write a program to count the sum of numbers in a string

## Code in C

```

#include<stdio.h>
int main()
{
    //Initializing variables.
    char str[100] = "4PREP2INSTA6";
    int i,sum = 0;

    //Iterating each character through for loop.
    for (i= 0; str[i] != '\0'; i++)
    {
        if ((str[i] >= '0') && (str[i] <= '9')) //Checking for numeric characters.

```

```

    {

        sum += (str[i] - '0'); //Adding numeric characters.

    }
}
//Printing result.
printf("Sum of all digits:\n%d", sum);
return 0;
}

```

## Code in C++

```

#include<iostream>
using namespace std;

int findSum(string str)
{
    int sum = 0;
    for (char ch : str)
    {
        if (isdigit(ch))
        {
            sum += ch - '0';
        }
    }
    return sum;
}

int main()
{
    string str="Pr22e44pinsta";
    cout << "Sum :" << findSum(str) << endl;
}

```

## Code in Java

```

import java.util.Scanner;
public class Main {
    public static void main(String[] args) {
        String s ="4PREP2INSTAA6";
        int sum=0;
        for (int i = 0; i < s.length(); i++) {
            if(Character.isDigit(s.charAt(i)))
                sum=sum+Character.getNumericValue(s.charAt(i));
        }
    }
}

```

```
    System.out.println("Sum of all the digit present in String : "+sum);  
  }  
}
```

### Code in Python

```
String = "Daya123Ben456"  
#initialize integer variable  
sum1 = 0  
for i in String:  
    #check if values lies between range of numbers or not  
    #according to ascii tale  
    if ord(i) >= 48 and ord(i) <= 57:  
        #convert it to integer and add  
        sum1 = sum1 + int(i)  
print('Sum is .' + str(sum1))
```

**Q69: Capitalize the first and last character of each word of a string**

Write a program to capitalize the first and last character of each word of a string

### Code in C

```
#include<stdio.h>
#include<ctype.h>
#include<string.h>
int main()
{
    //Initializing variables
    char str[100] = "str ing";
    int length = 0;

    //Calculating length.
    length = strlen(str);
    for(int i=0;i<length;i++)
    {
        if(i==0||i==(length-1)) //Converting character at first and last index to uppercase.
        {
            str[i]=toupper(str[i]);
        }
        else if(str[i]==' ') //Converting characters present before and after space to uppercase.
        {
            str[i-1]=toupper(str[i-1]);
            str[i+1]=toupper(str[i+1]);
        }
    }

    //Printing result.
    printf("String after capitalizing first and last letter of each word:\n%s", str);
    return 0;
}
```

### Code in C++

```
#include <iostream>
#include <ctype.h>
#include <string.h>
using namespace std;

int main()
{

    //Initializing variables.
    char str[100]="Prep insta";
```

```

int length = 0;
//Calculating length.
length = strlen(str);

for(int i=0;i<length;i++)
{
if(i==0||i==(length-1)) //Converting character at first and last index to uppercase.
{
str[i]=toupper(str[i]);
}
else if(str[i]==' ')//Converting characters present before and after space to uppercase.
{
str[i-1]=toupper(str[i-1]);
str[i+1]=toupper(str[i+1]);
}
}

cout<<"String after capitalizing first and last letter of each word:\n"<<str;
return 0;
}

```

## Code in Java

```

import java.util.Scanner;
public class Main {
    public static void main(String[] args) {
        Scanner sc =new Scanner(System.in);
        System.out.print("Enter String : ");
        String s = sc.nextLine();
        String newstr = "";

        String[] str = s.split("\\s"); // splitting sentence into word converted to String array

        for (String string : str) {
            int length = string.length();
            String firstchar = string.substring(0, 1);
            String restchar = string.substring(1, length - 1);
            String lastchar = Character.toString(string.charAt(length - 1));
            newstr = newstr+firstchar.toUpperCase()+restchar+" ";
        }
        System.out.println(newstr);
    }
}

```

## Code in Python

```
String = input('Enter the String :')
String = String[0:1].upper() + String[1:len(String)-1] + String[len(String)-1:len(String)].upper()
#print the String
print(String)
```

## Q70: Find the Union and Intersection of the two sorted arrays

Write a program to find the Union and Intersection of the two sorted arrays

## Code in C

```
#include <stdio.h>
void printUnion(int arr1[], int arr2[], int m, int n)
{
    int i = 0, j = 0;
    while (i < m && j < n) {
        if (arr1[i] < arr2[j])
            printf("%d ",arr1[i++]);

        else if (arr2[j] < arr1[i])
            printf("%d ",arr2[j++]);

        else {
            printf("%d ",arr2[j++]);
            i++;
        }
    }

    /* Print remaining elements of the larger array */
    while (i < m)
        printf("%d ",arr1[i++]);

    while (j < n)
        printf("%d ",arr2[j++]);
}

void printIntersection(int arr1[], int arr2[], int m, int n)
{
    int i = 0, j = 0;
    while (i < m && j < n) {
        if (arr1[i] < arr2[j])
            i++;
    }
}
```



```

        else if (arr2[j] < arr1[i])
            j++;
        else /* if arr1[i] == arr2[j] */
        {
            printf("%d ",arr2[j]);
            i++;
            j++;
        }
    }
}

/* Driver program to test above function */
int main()
{
    int m, n;

    scanf("%d", &m);

    int arr1[m];

    for(int i=0; i<m; i++) scanf("%d", &arr1[i]);

    scanf("%d", &n);

    int arr2[n];

    for(int i=0; i<n; i++) scanf("%d", &arr2[i]);

    // Function calling
    printf("Union : ");
    printUnion(arr1, arr2, m, n);

    printf("\nIntersection ");
    printIntersection(arr1, arr2, m, n);

    return 0;
}

```

## Code in C++

```

#include <bits/stdc++.h>
using namespace std;

/* Function prints union of arr1[] and arr2[]

```

```

m is the number of elements in arr1[]
n is the number of elements in arr2[] */
void printUnion(int arr1[], int arr2[], int m, int n)
{
    int i = 0, j = 0;
    while (i < m && j < n) {
        if (arr1[i] < arr2[j])
            cout << arr1[i++] << " ";

        else if (arr2[j] < arr1[i])
            cout << arr2[j++] << " ";

        else {
            cout << arr2[j++] << " ";
            i++;
        }
    }

    /* Print remaining elements of the larger array */
    while (i < m)
        cout << arr1[i++] << " ";

    while (j < n)
        cout << arr2[j++] << " ";
}

void printIntersection(int arr1[], int arr2[], int m, int n)
{
    int i = 0, j = 0;
    while (i < m && j < n) {
        if (arr1[i] < arr2[j])
            i++;
        else if (arr2[j] < arr1[i])
            j++;
        else /* if arr1[i] == arr2[j] */
        {
            cout << arr2[j] << " ";
            i++;
            j++;
        }
    }
}

/* Driver program to test above function */
int main()

```

```

{
    int m, n;
    cin>>m;

    int arr1[m];

    for(int i=0; i<m; i++)
        cin>>arr1[i];

    cin>>n;

    int arr2[n];

    for(int i=0; i<n; i++) cin>>arr2[i];

    // Function calling
    cout<<"Union : ";
    printUnion(arr1, arr2, m, n);

    cout<<"\nIntersection ";
    printIntersection(arr1, arr2, m, n);

    return 0;
}

```

## Code in Java

```

import java.util.*;
public
class Main {
    public
    static void Union(int[] arr1, int[] arr2) {
        int i = 0;
        int j = 0;

        while (i < arr1.length && j < arr2.length) {
            if (arr1[i] < arr2[j]) {
                System.out.print(arr1[i] + " ");
                i++;
            }
            else if (arr2[j] < arr1[i]) {

```

```

        System.out.print(arr2[j] + " ");
        j++;
    } else {
        System.out.print(arr1[i] + " ");

        i++;
        j++;
    }
}

while (i < arr1.length) {
    System.out.print(arr1[i] + " ");
    i++;
}
while (j < arr2.length) {
    System.out.print(arr2[j] + " ");
    j++;
}
}

public static void Intersection(int[] arr1, int[] arr2) {
    int i = 0;
    int j = 0;
    while (i < arr1.length && j < arr2.length) {
        if (arr1[i] < arr2[j]) {
            i++;
        } else if (arr1[i] > arr2[j]) {
            j++;
        } else {
            System.out.print(arr1[i] + " ");
            i++;
            j++;
        }
    }
}

public static void main(String[] args) throws Exception {
    Scanner scn = new Scanner(System.in);

    int n1 = 3;
    System.out.println("The size of the array 1:- " + n1);
    int[] arr1 = new int[] { 1, 2, 3 };
    System.out.println("The Elements of array 1");
    for (int i = 0; i < n1; i++) {
        System.out.print(arr1[i] + " ");
    }
}

```

```

System.out.println();

int n2 = 4;
System.out.println("The size of the array 2:- "+n2);

int[] arr2 = new int[]{4,5,6,7};
System.out.println("The Elements of array 2");

for (int i = 0; i < n2; i++) {
    System.out.print(arr2[i] + " ");
}
System.out.println();
System.out.println("The Union of the two array is ");
Union(arr1, arr2);
System.out.println();
Intersection(arr1, arr2);
}
}

```

## Code in Python

```

def Union(arr1, arr2):
    i = 0
    j = 0
    union_result = []

    while i < len(arr1) and j < len(arr2):
        if arr1[i] < arr2[j]:
            union_result.append(arr1[i])
            i += 1
        elif arr2[j] < arr1[i]:
            union_result.append(arr2[j])
            j += 1
        else:
            union_result.append(arr1[i])
            i += 1
            j += 1

    while i < len(arr1):
        union_result.append(arr1[i])
        i += 1
    while j < len(arr2):

```

```

        union_result.append(arr2[j])
        j += 1

    return union_result

def Intersection(arr1, arr2):
    i = 0
    j = 0
    intersection_result = []

    while i < len(arr1) and j < len(arr2):
        if arr1[i] < arr2[j]:
            i += 1
        elif arr2[j] < arr1[i]:
            j += 1
        else:
            intersection_result.append(arr1[i])
            i += 1
            j += 1

    return intersection_result

if __name__ == "__main__":
    n1 = 3
    print("The size of array 1:", n1)
    arr1 = [1, 2, 3]
    print("The elements of array 1:")
    print(" ".join(map(str, arr1)))

    n2 = 4
    print("The size of array 2:", n2)
    arr2 = [4, 5, 6, 7]
    print("The elements of array 2:")
    print(" ".join(map(str, arr2)))

    print("The Union of the two arrays is:")
    union_result = Union(arr1, arr2)
    print(" ".join(map(str, union_result)))

    print("The Intersection of the two arrays is:")
    intersection_result = Intersection(arr1, arr2)
    print(" ".join(map(str, intersection_result)))

```

## Q71: Calculate frequency of characters in a string

Write a Program to calculate frequency of characters in a string

### Code in C

```
#include<stdio.h>

int main()
{
    //Initializing variables.
    char str[100] ="prepinsta";
    int i;
    int freq[256] = {0};

    //Calculating frequency of each character.
    for(i = 0; str[i] != '\0'; i++)
    {
        freq[str[i]]++;
    }

    //Printing frequency of each character.
    for(i = 0; i < 256; i++)
    {
        if(freq[i] != 0)
        {
            printf("The frequency of %c is %d\n", i, freq[i]);
        }
    }
    return 0;
}
```

### Code in C++

```
#include <iostream>
using namespace std;

int main()
{
    //Initializing variables.
    char str[100]="prepinsta";
    int i;
    int freq[256] = {0};
```

```

//Calculating frequency of each character.
for(i = 0; str[i] != '\0'; i++)
{
    freq[str[i]]++;
}

//Printing frequency of each character.
for(i = 0; i < 256; i++)
{
    if(freq[i] != 0)
    {
        cout<<"The frequency of "<<char(i)<<" is "<<freq[i]<<endl;
    }
}
return 0;
}

```

## Code in Java

```

import java.util.Scanner;

public class Main {
    public static void main(String[] args) {
        Scanner sc =new Scanner(System.in);
        System.out.print("Enter String : ");
        String str = sc.nextLine();
        int[] freq = new int[str.length()];
        int i, j;

        //Converts given string into character array
        char string[] = str.toCharArray();
        for(i = 0; i <str.length(); i++) {
            freq[i] = 1;
            for(j = i+1; j <str.length(); j++) {
                if(string[i] == string[j]) {
                    freq[i]++;
                }

                //Set string[j] to 0 to avoid printing visited character
                string[j] = '0';
            }
        }

        //Displays the each character and their corresponding frequency
        System.out.println("Characters and their corresponding frequencies");
    }
}

```



```

for(i = 0; i < freq.length; i++) {
    if(string[i] != ' ' && string[i] != '0')
        System.out.println(string[i] + "-" + freq[i]);
    }
}
}

```

## Code in Python

```
string = "Yolo Life"
```

```

for i in string:
    frequency = string.count(i)
    print(str(i) + ": " + str(frequency), end=" ", ")

```

## Q72: Spiral traversal on a Matrix

Write a program Spiral traversal on a Matrix

### Code in C :

```

#include <stdio.h>
#define r 4
#define c 4

int main()
{
    int a[4][4] = { { 1, 2, 3, 4 },
                    { 5, 6, 7, 8 },
                    { 9, 10, 11, 12 },
                    { 13, 14, 15, 16 } };

    int i, left = 0, right = c-1, top = 0, bottom = r-1;

    while (left <= right && top <= bottom) {

        /* Print the first row
        from the remaining rows */
        for (i = left; i <= right; ++i) {
            printf("%d ", a[top][i]);
        }
    }
}

```

```

    top++;

    /* Print the last column
    from the remaining columns */
    for (i = top; i <= bottom; ++i) {
        printf("%d ", a[i][right]);
    }
    right--;

    /* Print the last row from
    the remaining rows */
    if (top <= bottom) { for (i = right; i >= left; --i) {
        printf("%d ", a[bottom][i]);
    }
    bottom--;
}

/* Print the first column from
the remaining columns */
if (left <= right) { for (i = bottom; i >= top; --i) {
    printf("%d ", a[i][left]);
}
left++;
}
}

return 0;
}

```

## Code in C++:

```

#include <bits/stdc++.h>
#define r 4
#define c 4
using namespace std;

int main()
{
    int a[4][4] = { { 1, 2, 3, 4 },
                    { 5, 6, 7, 8 },
                    { 9, 10, 11, 12 },

```

```

        { 13, 14, 15, 16 } };

int i, left = 0, right = c-1, top = 0, bottom = r-1;

while (left <= right && top <= bottom) {

    /* Print the first row
    from the remaining rows */
    for (i = left; i <= right; ++i) {
        cout<<a[top][i]<<" ";
    }
    top++;

    /* Print the last column
    from the remaining columns */
    for (i = top; i <= bottom; ++i) {
        cout<<a[i][right]<<" ";
    }
    right--;

    /* Print the last row from
    the remaining rows */
    if (top <= bottom) { for (i = right; i >= left; --i) {
        cout<<a[bottom][i]<<" ";
    }
    bottom--;
}

    /* Print the first column from
    the remaining columns */
    if (left <= right) { for (i = bottom; i >= top; --i) {
        cout<<a[i][left]<<" ";
    }
    left++;
}
}

return 0;
}

```

Code in Java:

```

import java.util.*;

class Main{

```

```
static int R = 4;
static int C = 4;
```

```
static void print(int arr[][], int i, int j, int m, int n)
{
    if (i >= m || j >= n) {
        return;
    }

    for (int p = i; p < n; p++) {
        System.out.print(arr[i][p] + " ");
    }

    for (int p = i + 1; p < m; p++) {
        System.out.print(arr[p][n - 1] + " ");
    }

    if ((m - 1) != i) {
        for (int p = n - 2; p >= j; p--) {
            System.out.print(arr[m - 1][p] + " ");
        }
    }

    if ((n - 1) != j) {
        for (int p = m - 2; p > i; p--) {
            System.out.print(arr[p][j] + " ");
        }
    }
    print(arr, i + 1, j + 1, m - 1, n - 1);
}
```

```
public static void main(String[] args)
{
    int a[][] = { { 1, 2, 3, 4 },
                  { 5, 6, 7, 8 },
                  { 9, 10, 11, 12 },
                  { 13, 14, 15, 16 } };

    print(a, 0, 0, R, C);
}
}
```

## Code in Python:

```
r = 4
```

```
c = 4
```

```
a = [[1, 2, 3, 4],  
      [5, 6, 7, 8],  
      [9, 10, 11, 12],  
      [13, 14, 15, 16]]
```

```
left = 0
```

```
right = c - 1
```

```
top = 0
```

```
bottom = r - 1
```

```
while left <= right and top <= bottom:
```

```
    """ Print the first row from the remaining rows"""
```

```
    for i in range(left, right + 1):
```

```
        print(a[top][i], end=" ")
```

```
    top += 1
```

```
    """ Print the last column from the remaining columns"""
```

```
    for i in range(top, bottom - 1, -1):
```

```
        print(a[i][right], end=" ")
```

```
    right -= 1
```

```
    """Print the last row from the remaining rows"""
```

```
    if top <= bottom:
```

```
        for i in range(right, left - 1, -1):
```

```
            print(a[bottom][i], end=" ")
```

```
        bottom -= 1
```

```
    """Print the first column from the remaining columns"""
```

```
    if left <= right:
```

```
        for i in range(bottom, top - 1, -1):
```

```
            print(a[i][left], end=" ")
```

```
        left += 1
```

## Q73: Find Largest sum contiguous Subarray

Write a program to find Largest sum contiguous Subarray

### Code in C

```
#include <stdio.h>

#define ARRAY_SIZE(a) sizeof(a)/sizeof(a[0])

int maxSubArraySum (int arr[], int n)
{
    int i = 0;
    int max_so_far = 0;
    int max_ending_here = 0;

    for (i = 0; i < n; i++)
    {
        max_ending_here = max_ending_here + arr[i];
        if (max_ending_here < 0)
        {
            max_ending_here = 0;
        }
        if (max_so_far < max_ending_here)
        {
            max_so_far = max_ending_here;
        }
    }
    return max_so_far;
}

int main ()
{
    int arr[] = { -2, 1, -3, 4, -1, 2, 1, -5, 4 };
    int arr_size = ARRAY_SIZE (arr);
    const int maxSum = maxSubArraySum (arr, arr_size);
    printf ("Largest Sum Contiguous Sub-Array : %d ", maxSum);
    return 0;
}
```

### Code in C++

```
#include <bits/stdc++.h>
using namespace std;
```

```

int main()
{
    int arr[] = {-2, -3, 4, -1, -2, 1, 5, -3};
    int n = sizeof(arr)/sizeof(arr[0]);

    int res = INT_MIN;

    for(int i=0; i<n; i++){
        int sum = 0;
        for(int j=i; j<n; j++){
            sum += arr[j];
            res = max(sum, res);
        }
    }

    cout<<res;
}

```

## Code in Java

```

import java.util.*;

public class Main {
    public static void main(String[] args) {
        int[] arr = {-2, -3, 4, -1, -2, 1, 5, -3};
        int n = arr.length;
        int res = Integer.MIN_VALUE;
        for (int i = 0; i < n; i++) {
            int sum = 0;
            for (int j = i; j < n; j++) {
                sum += arr[j];
                res = Math.max(sum, res);
            }
        }
        System.out.println(res);
    }
}

```

## Code in Python

```

def max_subarray_sum(arr):
    n = len(arr)
    max_sum = float('-inf')
    current_sum = 0

```

```

        for i in range(n):
            current_sum = max(arr[i], current_sum + arr[i])
            max_sum = max(max_sum, current_sum)

        return max_sum

if __name__ == "__main__":
    arr = [-2, -3, 4, -1, -2, 1, 5, -3]
    result = max_subarray_sum(arr)
    print(result)

```

## Q74: Find non-repeating characters in a string

Write a program to find non-repeating characters in a string

### Code in C

```

#include<stdio.h>
int main()
{
    //Initializing variables.
    char str[100]="prepinsta";
    int i;
    int freq[256] = {0};
    //Calculating frequency of each character.
    for(i = 0; str[i] != '\0'; i++)
    {
        freq[str[i]]++;
    }
    printf("The non repeating characters are: ");
    for(i = 0; i < 256; i++)
    {
        if(freq[i] == 1)//Finding uniques charcters and printing them.
        {
            printf(" %c ", i);
        }
    }
    return 0;
}

```



## Code in C++

```
#include <iostream>
using namespace std;
int main()
{
    //Initializing variables.
    char str[100]="prepinsta";
    int i;
    int freq[256] = {0};

    //Calculating frequency of each character.
    for(i = 0; str[i] != '\0'; i++)
    {
        freq[str[i]]++;
    }
    cout<<"The non repeating characters are: ";
    for(i = 0; i < 256; i++)
    {
        if(freq[i] == 1)//Finding non repeating charcters and printing them.
        {
            cout<<char(i)<<" ";
        }
    }
    return 0;
}
```

## Code in Java

```
class Main
{
    public static void main(String args[])
    {
        String inputStr ="prepinsta";
        boolean flag = true;

        for(char i :inputStr.toCharArray())
        {
            // if current character is the last occurrence in the string
            if (inputStr.indexOf(i) == inputStr.lastIndexOf(i))
            {
                System.out.println("First non-repeating character is: "+ i);
                flag = false;
                break;
            }
        }
    }
}
```

```

    }

    if(flag)
        System.out.println("There is no non repeating character in input string");
    }
}

```

## Code in Python

```

String = "prepinsta"
for i in String:
    #initialize a count variable
    count = 0
    for j in String:
        #check for repeated characters
        if i == j:
            count+=1

    if count > 1:
        break
#print for nonrepeating characters
if count == 1:
    print(i,end = " ")

```

## Q75: Minimize the maximum difference between heights

Write a program to minimize the maximum difference between heights

## Code in C

```

#include <stdio.h>

int min(int a, int b){
    if(a>b)
        return b;
    return a;
}

int max(int a, int b){
    if(a<b)
        return b;
    return a;
}

```

```

int getMinDiff(int arr[], int n, int k)
{
    //Sort the array
    for(int i=0; i<n; i++){
        for(int j=i+1; j<n; j++){ if(arr[i]>arr[j]){
            int temp = arr[i];
            arr[i] = arr[j];
            arr[j] = temp;
        }
    }
}

int ans = arr[n - 1] - arr[0];

int tempmin, tempmax;
tempmin = arr[0];
tempmax = arr[n - 1];

for (int i = 1; i < n; i++) {
    tempmin= min(arr[0] + k,arr[i] - k);
    tempmax = max(arr[i - 1] + k, arr[n - 1] - k);
    ans = min(ans, tempmax - tempmin);
}
return ans;
}

// Driver Code Starts
int main()
{
    int k = 6, n = 6;
    int arr[] = { 7, 4, 8, 8, 8, 9 };
    int ans = getMinDiff(arr, n, k);

    printf("%d", ans);
}

```

## Code in C++

```

#include<bits/stdc++.h>
using namespace std;

int getMinDiff(int arr[], int n, int k){

    if (n == 1)

```

```

return 0;

sort(arr, arr + n);

vector<pair<int, int>> t;

map<int, int> m;

int n_ = 1;

t.push_back(pair<int, int>(arr[0] + k, arr[0]));

t.push_back(pair<int, int>(arr[0] - k, arr[0]));

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

    if (arr[i] != arr[i - 1]) {

        t.push_back(pair<int, int>(arr[i] + k, arr[i]));

        t.push_back(pair<int, int>(arr[i] - k, arr[i]));

        m[arr[i]] = 0;

        n_++;
    }
}

sort(t.begin(), t.end());

int l = 0, r = 0;

int ans = t[t.size() - 1].first - t[0].first;

int count = 0;

while (r < t.size()) {

    while (r < t.size() and count < n_) {

        if (m[t[r].second] == 0)
            count++;

        m[t[r].second]++;
    }
}

```

```

        r++;
    }

    if (r == t.size() and count < n_)
        break;

    ans = min(ans, t[r - 1].first - t[l].first);

    while (l <= r and count >= n_) {

        if (m[t[l].second] == 1)
            count--;

        m[t[l].second]--;

        ans = min(ans, t[r - 1].first - t[l].first);

        l++;
    }
}
return ans;
}

int main()
{
    int arr[] = {1, 10, 14, 14, 14, 15};
    int n = sizeof(arr)/sizeof(arr[0]);

    int k = 6;

    cout << getMinDiff(arr, n, k);

}

```

## Code in Java

```

import java.io.*;
import java.util.*;

public class Main {

    public static void main(String[] args)
    {
        int[] arr = { 7, 4, 8, 8, 8, 9 };
    }
}

```

```

        int k = 6;
        int ans = getMinDiff(arr, arr.length, k);
        System.out.println(ans);
    }
// } Driver Code Ends

// User function Template for Java

public static int getMinDiff(int[] arr, int n, int k)
{
    Arrays.sort(arr);
    int ans = (arr[n - 1] + k) - (arr[0] + k); // Maximum possible height difference

    int tempmax = arr[n - 1] - k; // Maximum element when we subtract k from whole array
    int tempmin = arr[0] + k; // Minimum element when we add k to whole array
    int max, min;

    for (int i = 0; i < n - 1; i++) {
        if (tempmax > (arr[i] + k)) {
            max = tempmax;
        }
        else {
            max = arr[i] + k;
        }

        if (tempmin < (arr[i + 1] - k)) {
            min = tempmin;
        }
        else {
            min = arr[i + 1] - k;
        }

        if (ans > (max - min)) {
            ans = max - min;
        }
    }
    return ans;
}
}

```

## Code in Python

```
def profit(arr, k):
    n = (min(arr) + max(arr)) // 2
    new = []
    for i in arr:
        if max(arr) - min(arr) < k:
            return max(arr) - min(arr)
        elif i >= n:
            new.append(i - k)
        else:
            new.append(i + k)
    return max(new) - min(new)
```

```
array = [2, 9, 16]
K = 6
print("Maximum difference is :", profit(array, K))
```

## Q76: Check if two strings are Anagram or not

Write a program to check if two strings are Anagram or not

## Code in C

```
#include<stdio.h>
int main()
{
    //Initializing variables.
    char str1[100]="prep",str2[100]="perp";
    int first[26]={0}, second[26]={0}, c=0, flag=0;

    //Calculating frequencies of characters in first string.
    while(str1[c] != '\0')
    {
        first[str1[c]-'a']++;
        c++;
    }

    c=0;
    //Calculating frequencies of characters in second string.
    while(str2[c] != '\0')
```

```

{
    second[str2[c]-'a']++;
    c++;
}
//Checking if frequencies of both the strings are same or not.
for(c=0;c<26;c++)
{
    if(first[c] != second[c])
        flag=1;
}
//Printing result.
if(flag == 0)
{
    printf("\n%s and %s are Anagram Strings.",str1,str2);
}
else
{
    printf("\n%s and %s are not Anagram Strings.",str1,str2);
}
return 0;
}

```

## Code in C++

```

#include<iostream>
using namespace std;
int main()
{
    //Initializing variables.
    char str1[100],str2[100];
    int first[26]={0}, second[26]={0}, c=0, flag=0;

    //Accepting inputs.
    cout<<"Enter First String: ";
    cin>>str1;
    cout<<"Enter Second String: ";
    cin>>str2;

    //Calculating frequencies of characters in first string.
    while(str1[c] != '\0')
    {
        first[str1[c]-'a']++;
        c++;
    }
}

```



```

c=0;
//Calculating frequencies of characters in second string.
while(str2[c] != '\0')
{
    second[str2[c]-'a']++;
    c++;
}
//Checking if frequencies of both the strings are same or not.
for(c=0;c<26;c++)
{
    if(first[c] != second[c])
        flag=1;
}
//Printing result.
if(flag == 0)
{
    cout<<"Strings are anagram.";
}
else
{
    cout<<"Strings are not anagram.";
}
return 0;
}

```

## Code in Java

```

import java.util.Arrays;
import java.util.Scanner;
public class Main {
    static boolean isAnagram(String str1 , String str2) {
        String s1 = str1.replaceAll("[\\s]", "");
        String s2 = str2.replaceAll("[\\s]", "");
        boolean status=true;

        if(s1.length()!=s2.length())
            status = false;
        else {
            char[] a1 = s1.toLowerCase().toCharArray();
            char[] a2 = s2.toLowerCase().toCharArray();
            Arrays.sort(a1);
            Arrays.sort(a2);
            status = Arrays.equals(a1, a2);
        }
    }
}

```

```

    }
    return status;
}

public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    System.out.print("Enter two String :");
    String s1 = sc.next();
    String s2 = sc.next();
    boolean status = isAnagram(s1,s2);
    if(status)
        System.out.println(s1+" and "+s2+" are Anagram");
    else
        System.out.println(s1+" and "+s2+" are not Anagram");
    }
}

```

## Code in Python

```

String1 = "Listen"
String2 = "Silent"

String1 = sorted(String1.lower())
String2 = sorted(String2.lower())

print("String1 after sorting: ", String1)
print("String2 after sorting: ", String2)

# check if now strings matches
if String1 == String2:
    print('Strings are anagram')
else:
    print('Strings are not anagram')

```

## Q77: Minimum no. of Jumps to reach the end of an array

Write the program to find the minimum no. of Jumps to reach the end of an array

## Code in C

```

#include <limits.h>
#include <stdio.h>

int min(int x, int y) { return (x < y) ? x : y; }

```

```

int minJumps(int arr[], int n)
{
    // jumps[n-1] will hold the result
    int jumps[n];
    int i, j;

    if (n == 0 || arr[0] == 0)
        return INT_MAX;

    jumps[0] = 0;

    // Find the minimum number of jumps to reach arr[i]
    // from arr[0], and assign this value to jumps[i]
    for (i = 1; i < n; i++) {
        jumps[i] = INT_MAX;
        for (j = 0; j < i; j++) {
            if (i <= j + arr[j] && jumps[j] != INT_MAX) {
                jumps[i] = min(jumps[i], jumps[j] + 1);
                break;
            }
        }
    }
    return jumps[n - 1];
}

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

    int arr[n];
    for(int i=0; i<n; i++)
        scanf("%d", &arr[i]);

    printf("Minimum number of jumps to reach end is %d ",minJumps(arr, n));

    return 0;
}

```

## Code in C++

```

#include <bits/stdc++.h>
using namespace std;

```

```

int minJumps(int arr[], int n)
{
    // jumps[n-1] will hold the result
    int jumps[n];
    int i, j;

    if (n == 0 || arr[0] == 0)
        return INT_MAX;

    jumps[0] = 0;

    // Find the minimum number of jumps to reach arr[i]
    // from arr[0], and assign this value to jumps[i]
    for (i = 1; i < n; i++) {
        jumps[i] = INT_MAX;
        for (j = 0; j < i; j++) {
            if (i <= j + arr[j] && jumps[j] != INT_MAX)
            {
                jumps[i] = min(jumps[i], jumps[j] + 1);
                break;
            }
        }
    }
    return jumps[n - 1];
}

int main()
{
    int n;
    cin>>n;
    int arr[n];

    for(int i=0; i<n; i++) cin>>arr[i];

    cout<<"Minimum number of jumps to reach end is "<< minJumps(arr, n);

    return 0;
}

```

## Code in Java

```

public class Main {

```

```

private static int minJumps(int[] arr, int n) {
// jumps[n-1] will hold the
int jumps[] = new int[n];

int i, j;
// if first element is 0,
if (n == 0 || arr[0] == 0) return Integer.MAX_VALUE;

// end cannot be reached
jumps[0] = 0;
// Find the minimum number of jumps to reach arr[i]
// from arr[0], and assign this value to jumps[i]

for (i = 1; i < n; i++) {
jumps[i] = Integer.MAX_VALUE;

for (j = 0; j < i; j++) {
    if (i <= j + arr[j] && jumps[j] != Integer.MAX_VALUE) {
        jumps[i] = Math.min(jumps[i], jumps[j] + 1);
        break;
    }
}
}
return jumps[n - 1];
}

// driver program to test above function
public static void main(String[] args) {
int arr[] = {2, 1, 3, 2, 3, 4, 5, 1, 2, 8};

System.out.println("Minimum number of jumps to reach end is : " +
    minJumps(arr, arr.length));
}
}

```

## Code in Python

```

def jump(arr):
    ans = 0
    i = 0
    while i < len(arr) - 1:
        if i + arr[i] < len(arr):
            ans += 1
            if arr[i] == 1:
                i += arr[i]

```

```

        else:
            i += arr.index(max(arr[i + 1:arr[i] + i + 1])) - i
    else:
        ans += 1
        i += arr[i]

```

```

return ans

```

```

arr = [1, 3, 5, 8, 9, 2, 6, 7, 6, 8, 9]
print("Minimum no of jumps required to reach end of the array : ", jump(arr))

```

## Q78: Replace a sub-string in a string

Write a program to replace a sub-string in a string

Code in C

```

#include<stdio.h>
#include<string.h>
int main() {
    char str[256] = "prepinsta", substr[128] = "insta", replace[128] = "ster ", output[256];
    int i = 0, j = 0, flag = 0, start = 0;

    str[strlen(str) - 1] = '\0';
    substr[strlen(substr) - 1] = '\0';
    replace[strlen(replace) - 1] = '\0';

    // check whether the substring to be replaced is present
    while (str[i] != '\0')
    {
        if (str[i] == substr[j])
        {
            if (!flag)
                start = i;
            j++;
            if (substr[j] == '\0')
                break;
            flag = 1;
        }
        else
        {

```

```

        flag = start = j = 0;
    }
    i++;
}
if (substr[j] == '\0' && flag)
{
    for (i = 0; i < start; i++)
        output[i] = str[i];

    // replace substring with another string
    for (j = 0; j < strlen(replace); j++)
    {
        output[i] = replace[j];
        i++;
    }
    // copy remaining portion of the input string "str"
    for (j = start + strlen(substr); j < strlen(str); j++)
    {
        output[i] = str[j];
        i++;
    }
    // print the final string
    output[i] = '\0';
    printf("Output: %s\n", output);
} else {
    printf("%s is not a substring of %s\n", substr, str);
}
return 0;
}

```

## Code in C++

```

#include<iostream>
#include<string.h>
using namespace std;
void replaceSubstring(char st[],char sub[],char new_str[])//Function to replace substring.
{
    int stLen,subLen,newLen;
    int i=0,j,k;
    int flag=0,start,end;
    stLen=strlen(st);
    subLen=strlen(sub);
    newLen=strlen(new_str);
    for(i=0;i<stLen;i++)//Finding substring.
    {

```

```

    flag=0;
    start=i;
    for(j=0;st[i]==sub[j];j++,i++)
        if(j==subLen-1)
            flag=1;
    end=i;
    if(flag==0)
        i=j;
    else
    {
        for(j=start;j<end;j++)
        {
            for(k=start;k<stLen;k++)
                st[k]=st[k+1];
            stLen--;
            i--;
        }
        for(j=start;j<start+newLen;j++)//Replacing sub string with the input string
        {
            for(k=stLen;k>=j;k--)
                st[k+1]=st[k];
            st[j]=new_str[j-start];
            stLen++;
            i++;
        }
    }
}
}
//Main function.
int main()
{
    char st[100] = "prepinsta",sub[100] = "insta",new_str[100]="ster ";

    replaceSubstring(st,sub,new_str); //Calling created function.
    //Printing result using called function.
    cout<<"The string after replacing substring: "<<st<<endl;
    return 0;
}

```

## Code in Java

```

public class Main {
    public static void main(String[] args) {
        String originalString = "Hello, World!";
        String substringToReplace = "World";
    }
}

```



```

String replacement = "Java";

// Replace the substring
String modifiedString = replaceSubstring(originalString, substringToReplace, replacement);

// Print the modified string
System.out.println("Original String: " + originalString);
System.out.println("Modified String: " + modifiedString);
}

public static String replaceSubstring(String original, String toReplace, String replacement) {
    // Check if the original string contains the substring to replace
    if (!original.contains(toReplace)) {
        return original; // If not, return the original string as it is
    }

    // Use the replaceAll method to replace all occurrences of the substring
    String modified = original.replaceAll(toReplace, replacement);
    return modified;
}
}

```

## Code in Python

```

string = input("Enter String:\n")
str1 = input("Enter substring which has to be replaced:\n") # User inputs the substring to be replaced
str2 = input("Enter substring with which str1 has to be replaced:\n") # User inputs the replacement substring

string = string.replace(str1, str2) # Replace all occurrences of str1 with str2 in the input string

print("String after replacement:")
print(string) # Print the updated string

```

## Q79: Replacing a particular word with another word in a string

Write a program to replacing a particular word with another word in a string

## Code in C

```

#include <stdio.h>

```

```

#include <string.h>
#include <stdlib.h>

// Function to replace a string with another string
char* replaceWord(const char* s, const char* old_word, const char* new_word)
{
    char* result;
    int i, cnt = 0;
    int newWlen = strlen(new_word);
    int oldWlen = strlen(old_word);

    // Counting the number of times old word
    // occur in the string
    for (i = 0; s[i] != '\0'; i++) {
        if (strstr(&s[i], old_word) == &s[i]) {
            cnt++;

            // Jumping to index after the old word.
            i += oldWlen - 1;
        }
    }

    // Making new string of enough length
    result = (char*)malloc(i + cnt * (newWlen - oldWlen) + 1);

    i = 0;
    while (*s) {
        // compare the substring with the result
        if (strstr(s, old_word) == s) {
            strcpy(&result[i], new_word);
            i += newWlen;
            s += oldWlen;
        }
        else
            result[i++] = *s++;
    }

    result[i] = '\0';
    return result;
}

// Driver Program
int main()
{

```

```

char str[] = "Let's Learn C++";
char c[] = "C++";
char d[] = "C";

char* result = NULL;

// oldW string
result = replaceWord(str, c, d);
printf("Old String: %s\n", str);
printf("New String: %s\n", result);

return 0;
}

```

## Code in C++

```

#include <bits/stdc++.h>
using namespace std;

// Function to replace all the occurrences
// of the substring S1 to S2 in string S
void modifyString(string& s, string& s1,
                  string& s2)
{
    // Stores the resultant string
    string ans = "";

    // Traverse the string s
    for (int i = 0; i < s.length(); i++) {

        int k = 0;

        // If the first character of
        // string s1 matches with the
        // current character in string s
        if (s[i] == s1[k]
            && i + s1.length()
            <= s.length()) {

            int j;

            // If the complete string
            // matches or not
            for (j = i; j < i + s1.length(); j++) {

```

```

        if (s[j] != s1[k]) {
            break;
        }
        else {
            k = k + 1;
        }
    }

    // If complete string matches
    // then replace it with the
    // string s2
    if (j == i + s1.length()) {
        ans.append(s2);
        i = j - 1;
    }

    // Otherwise
    else {
        ans.push_back(s[i]);
    }
}

// Otherwise
else {
    ans.push_back(s[i]);
}
}

// Print the resultant string
cout << "Modified String : " << ans;
}

// Driver Code
int main()
{
    string S = "Let's Learn C";
    string S1 = "C";
    string S2 = "C++";
    cout << "Original String : " << S << endl;
    modifyString(S, S1, S2);

    return 0;
}

```

## Code in Java

```
public
class Main {
    static void remove(String str, String word) {
        String msg[] = str.split(" ");
        String new_str = "";

        // Iterating the string using for each loop
        for (String words : msg) {

            // If desired word is found
            if (!words.equals(word)) {
                // Concat the word not equal to the given word
                new_str += words + " ";
            }
        }
        // Print the new String
        System.out.print(new_str);
    }
    public
    static void main(String[] args) {
        // Custom string as input
        String str = "This is the prepinsta";
        // Word to be removed from above string
        String word = "the";
        // Calling the method 1 by passing both strings to it
        remove(str, word);
    }
}
```

## Code in Python

```
def remove_word(input_str, word_to_remove):
    words = input_str.split()
    new_str = []

    for word in words:
        if word != word_to_remove:
            new_str.append(word)

    result = ' '.join(new_str)
    print(result)

def main():
```

```
input_str = "This is the prepinsta"
word_to_remove = "the"

remove_word(input_str, word_to_remove)

if __name__ == "__main__":
    main()
```

## Q80: Find duplicate in an array of N+1 Integers

Write a program to find duplicate in an array of N+1 Integers

### Code in C

```
#include<stdio.h>

int main()
{

    int n;
    scanf("%d", &n);

    int arr[n];

    for(int i=0; i < n; i++)
        scanf("%d", &arr[i]);

    int temp[n+1];
    for(int i=0; i <= n; i++)
        temp[i]=0;

    for(int i=0; i < n; i++)
        temp[arr[i]]++;

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

## Code in C++

```
#include <bits/stdc++.h>
using namespace std;

int main ()
{

    int n;
    cin >> n;

    int arr[n];

    for (int i = 0; i < n; i++)
        cin >> arr[i];

    map < int, int > mp;

    for (int i = 0; i < n; i++)
        mp[arr[i]]++;

    for (auto it = mp.begin (); it != mp.end (); it++)
    {
        if (it->second > 1)
            cout << it->first << " ";
    }
    return 0;
}
```

## Code in Java

```
import java.io.*;
import java.util.*;
public class Main {
    static int findduplicate(int[] arr, int n) {

        // return -1 because in these cases
        // there can not be any repeated element

        if (n <= 1) return -1;
```

```

// initialize fast and slow
int slow = arr[0];
int fast = arr[arr[0]];

// loop to enter in the cycle
while (fast != slow) {
// move one step for slow
slow = arr[slow];
// move two step for fast
fast = arr[arr[fast]];
}
// loop to find entry point of the cycle
fast = 0;
while (slow != fast) {
slow = arr[slow];
fast = arr[fast];
}
return slow;
}
// Driver Code

public static void main(String args[]) {
int[] arr = {1, 2, 3, 4, 5, 6, 3};
int n = arr.length;
System.out.print(findduplicate(arr, n));
}
}

```

## Code in Python

```

def find_duplicate(arr):
    n = len(arr)

    if n <= 1:
        return -1 # No repeated element when there's only one element or none

    # Initialize fast and slow pointers
    slow = arr[0]
    fast = arr[arr[0]]

    # Loop to enter the cycle
    while fast != slow:
        slow = arr[slow]
        fast = arr[arr[fast]]

```



```

        # Loop to find the entry point of the cycle
        fast = 0
        while slow != fast:
            slow = arr[slow]
            fast = arr[fast]

        return slow

if __name__ == "__main__":
    arr = [1, 2, 3, 4, 5, 6, 3]
    n = len(arr)
    print(find_duplicate(arr))

```

## Q81: Search an element in a matrix

Write a program to search an element in a matrix

### Code in C

```

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

    int a[4][4] = {{0,1,12,3}, {4,5,6,7}, {8,9,10,11}};;

    int search = 6;
    int flag=0;

    for(int i=0; i<4; i++){

        for(int j=0; j<4; j++){
            if(a[i][j]==search){
                printf("Element is found at (%d, %d) position",i,j);
                flag=1;
                break;
            }
        }
    }

    if(flag==1)
        break;
}

```

```
    if(flag==0)
        printf("Not found");
}
```

## Code in C++

```
#include<bits/stdc++.h>
using namespace std;
```

```
int main(){

    int n;
    cin>>n;

    int a[n][n];
    for(int i=0; i< n; i++){
        for(int j=0; j< n; j++)
            cin>>a[i][j];
    }

    int search;
    cin>>search;

    bool flag=0;

    for(int i=0; i< n; i++){

        for(int j=0; j< n; j++){
            if(a[i][j]==search)
            {
                cout<<"Element is found at ("<< i<<" , "<< j<<" ) position";
                flag=1;
                break;
            }
        }

        if(flag==1)
            break;
    }

    if(flag==0)
        cout<<"Not found";
}
```

## Code in Java

```
import java.io.*;
import java.util.*;

class Main {
    public static void main(String[] args)
    {
        int a[][] = {{0,1,12,3}, {4,5,6,7}, {8,9,10,11}};;

        int size = 4;
        int search = 6;
        int flag=0;

        for(int i=0; i<size; i++){

            for(int j=0; j<size; j++){
                if(a[i][j]==search){
                    System.out.println("Element is found at (" + i + ", " + j + ") position");
                    flag=1;
                    break;
                }
            }

            if(flag==1)
                break;
        }

        if(flag==0)
            System.out.println("Not found");
    }
}
```

## Code in Python

```
def main():
    a = [[0, 1, 12, 3], [4, 5, 6, 7], [8, 9, 10, 11]]
    size = 4
    search = 6
    flag = 0

    for i in range(size):
        for j in range(size):
            if a[i][j] == search:
                print(f"Element is found at ({i}, {j}) position")
```

```

        flag = 1
        break

    if flag == 1:
        break

    if flag == 0:
        print("Not found")

if __name__ == "__main__":
    main()

```

## Q82: Merge 2 sorted arrays without using extra space.

Write a program to Merge 2 sorted arrays without using extra space.

### Code in C

```

#include <stdio.h>

void merge(int arr1[], int arr2[], int m, int n)
{
    // Iterate through all elements
    // of arr2[] starting from the last element
    for (int i = n - 1; i >= 0; i--)
    {
        /* Find the smallest element greater than arr2[i].
        Move all elements one position ahead till the
        smallest greater element is not found */
        int j, last = arr1[m - 1];

        for (j = m - 2; j >= 0 && arr1[j] > arr2[i]; j--)
            arr1[j + 1] = arr1[j];

        // If there was a greater element
        if (j != m - 2 || last > arr2[i])
        {
            arr1[j + 1] = arr2[i];
            arr2[i] = last;
        }
    }
}

```

```

// Driver program
int main()
{

    int m, n;

    scanf("%d", &m);

    int arr1[m];

    for(int i=0; i<m; i++) scanf("%d", &arr1[i]);

    scanf("%d", &n);

    int arr2[n];

    for(int i=0; i<n; i++) scanf("%d", &arr2[i]);

    merge(arr1, arr2, m, n);

    printf("After Merging nFirst Array: ");

    for (int i = 0; i < m; i++)
        printf("%d ", arr1[i] );

    printf( "\nSecond Array: ");

    for (int i = 0; i < n; i++)
        printf("%d ", arr1[i] );

    return 0;
}

```

## Code in C++

```

#include <bits/stdc++.h>
using namespace std;

void merge(int arr1[], int arr2[], int m, int n)
{
    // Iterate through all elements
    // of ar2[] starting from the last element

```

```

for (int i = n - 1; i >= 0; i--)
{
    /* Find the smallest element greater than arr2[i].
    Move all elements one position ahead till the
    smallest greater element is not found */
    int j, last = arr1[m - 1];

    for (j = m - 2; j >= 0 && arr1[j] > arr2[i]; j--)
        arr1[j + 1] = arr1[j];

    // If there was a greater element
    if (j != m - 2 || last > arr2[i])
    {
        arr1[j + 1] = arr2[i];
        arr2[i] = last;
    }
}
}

```

```

// Driver program
int main()
{
    int m, n;
    cin >> m;

    int arr1[m];

    for(int i=0; i<m; i++) cin >> arr1[i];

    cin >> n;

    int arr2[n];

    for(int i=0; i<n; i++) cin >> arr2[i];

    merge(arr1, arr2, m, n);

    cout << "After Merging nFirst Array: ";

    for (int i = 0; i < m; i++)
        cout << arr1[i] << " ";

    cout << "\nSecond Array: ";

```

```

for (int i = 0; i < n; i++)
    cout << arr2[i] << " ";

return 0;
}

```

## Code in Java

```

import java.util.Arrays;

public class Main {
    static int arr1[] = new int[]{1, 12, 9, 3, 17, 20};
    static int arr2[] = new int[]{2, 3, 8, 13};
    static void merge(int m, int n) {
        // Iterate through all elements of arr2[] starting from
        // the last element
        for (int i = n - 1; i >= 0; i--) {
            int j, last = arr1[m - 1];
            for (j = m - 2; j >= 0 && arr1[j] > arr2[i]; j--) arr1[j + 1] = arr1[j];
            // If there was a greater element
            if (j != m - 2 || last > arr2[i]) {
                arr1[j + 1] = arr2[i];
                arr2[i] = last;
            }
        }
    }
    // Driver method to test the above function
    public static void main(String[] args) {
        merge(arr1.length, arr2.length);
        System.out.print("After Merging First Array: ");
        System.out.println(Arrays.toString(arr1));
        System.out.print("Second Array: ");
        System.out.println(Arrays.toString(arr2));
    }
}

```

## Code in Python

```

def find(array1, array2, n1, n2):
    # append array2 to array1
    for i in array2:
        array1.append(i)
    array1 = list(set(sorted(array1)))

    array2 = array1[len(array1) - n2:]

```

```

array1 = array1[:len(array1) - n2]

print("After")
print("Array1: ", array1, "\nArray2: ", array2)

```

```

array1 = [1, 2, 3, 5, 8, 9, 10, 13, 15, 20]
array2 = [2, 3, 8, 13]

```

```

print("Before: ")
print("Array1: ", array1)
print("Array2: ", array2)

```

```

find(array1, array2, len(array1), len(array2))

```

## Q83: Find median in a row wise sorted matrix

Write a program to find median in a row wise sorted matrix

### Code in C

```

#include <stdio.h>

int main(){
    int mat[3][3] ={{1, 3, 5},
                    {2, 6, 9},
                    {3, 6, 9}};
    int arr[9], x=0;
    for(int i=0; i<3; i++){
        for(int j=0; j<3; j++){
            arr[x++] = mat[i][j];
        }
    }

    for(int i=0; i<9; i++){
        for(int j=i+1; j<9; j++){
            if(arr[i]>arr[j]){
                int temp = arr[i];
                arr[i] = arr[j];
                arr[j] = temp;
            }
        }
    }
}

```



```
    printf("Median of the given matrix is : %d", arr[4]);  
}
```

## Code in C++

```
#include<bits/stdc++.h>  
using namespace std;  
  
int main(){  
    int mat[3][3] = { {1, 3, 5}, {2, 6, 9}, {3, 6, 9}};  
    int arr[9], x=0;  
    for(int i=0; i<3; i++){  
        for(int j=0; j<3; j++){  
            arr[x++] = mat[i][j];  
        }  
    }  
  
    sort(arr, arr+9);  
    cout << "Median of the given matrix is : " << arr[4];  
}
```

## Code in Java

```
import java.util.Arrays;  
  
public class Main  
{  
  
    public static void main(String[] args)  
    {  
        int r = 3, c = 3;  
        int mat[][]= { {1,3,5}, {2,6,9}, {3,6,9} };  
  
        int[] arr;  
  
        arr = new int[9];  
        int x=0;  
  
        for(int i=0; i<3; i++){  
            for(int j=0; j<3; j++){  
                arr[x++] = mat[i][j];  
            }  
        }  
    }  
}
```

```

        for(int i=0; i<9; i++){
            for(int j=i+1; j<9; j++){
                if(arr[i]>arr[j]){
                    int temp = arr[i];
                    arr[i] = arr[j];
                    arr[j] = temp;
                }
            }
        }
    }

    System.out.println("Median of the given matrix is : "+ arr[4]);

}
}

```

## Code in Python

```

mat = [[1, 3, 5],
        [2, 6, 9],
        [3, 6, 9]]

arr = []

for i in range(3):
    for j in range(3):
        arr.append(mat[i][j])

arr.sort()

print("Median of the given matrix is :", arr[4])

```

## Q84: Kadane's Algorithm

Write a program to explain kadane's Algorithm

## Code in C

```

#include <stdio.h>
#include <limits.h>

int main()
{

```

```

int n;
scanf("%d", &n);

int arr[n];

for(int i=0; i<n; i++)
    scanf("%d", &arr[i]);

int max_sum = INT_MIN, curr_sum =0;

for(int i=0; i<n; i++){

    curr_sum += arr[i];

    if(max_sum < curr_sum)
        max_sum = curr_sum;

    if(curr_sum < 0)
        curr_sum = 0;

}

printf("%d ", max_sum);

return 0;
}

```

## Code in C++

```

#include <bits/stdc++.h>
using namespace std;

int main()
{
    int n;
    cin>>n;

    int arr[n];

    for(int i=0; i<n; i++)
        cin>>arr[i];

    int max_sum = INT_MIN, curr_sum =0;

    for(int i=0; i<n; i++){

```

```

curr_sum += arr[i];

if(max_sum < curr_sum)
    max_sum = curr_sum;

if(curr_sum < 0)
    curr_sum = 0;

}

cout<<max_sum;

return 0;
}

```

## Code in Java

```

public class Main {
    public static void main(String[] args) {
        int[] a = {-2, -3, 4, -1, -3};
        System.out.println("Maximum contiguous sum is " + maxSubArraySum(a));
    }
    static int maxSubArraySum(int a[]) {
        int size = a.length;
        int max_so_far = Integer.MIN_VALUE, max_ending_here = 0;
        for (int i = 0; i < size; i++) {
            max_ending_here = max_ending_here + a[i];
            if (max_so_far < max_ending_here) max_so_far = max_ending_here;
            if (max_ending_here < 0) max_ending_here = 0;
        }
        return max_so_far;
    }
}

```

## Code in Python

```

def fun(arr, l):
    max_so_far = max(arr)
    for i in range(l - 1):
        s = arr[i]
        for j in range(i + 1, l):
            s += arr[j]
        if s > max_so_far:
            max_so_far = s

```

```
return max_so_far
```

```
array = [-2, -3, 4, -1, -2, 1, 5, -3]
```

```
print("Largest contiguous subarray sum is :", fun(array, len(array)))
```

## Q85: Find row with maximum no. of 1's

Write a program to find row with maximum no. of 1's

### Code in C

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

```
int main(){
```

```
    int mat[4][4] = {{0, 0, 0, 1},
                     {0, 1, 1, 1},
                     {1, 1, 1, 1},
                     {0, 0, 0, 0}};
```

```
    int max_count=0, index=-1;
```

```
    for(int i=0; i<4; i++){
        int count = 0;
        for(int j=0; j<4; j++){
            if(mat[i][j]==1)
                count++;
```

```
        }
        if(count>max_count)
        {
            max_count = count;
            index = i;
        }
    }
```

```
    printf("Index of row with maximum 1s is %d", index);
```

```
}
```

## Code in C++

```
#include<bits/stdc++.h>
using namespace std;

int main(){

    bool mat[4][4] = { {0, 0, 0, 1},
                       {0, 1, 1, 1},
                       {1, 1, 1, 1},
                       {0, 0, 0, 0}};

    int max_count=0, index=-1;

    for(int i=0; i<4; i++){
        int count = 0;
        for(int j=0; j<4; j++){
            if(mat[i][j]==1) count++;
        }
        if(count>max_count){
            max_count = count;
            index = i;
        }
    }

    cout << "Index of row with maximum 1s is " << index;

}
```

## Code in Java

```
public class Main {

    static boolean isPerfectSquare(int x)
    {
        if (x >= 0) {

            int sr = (int)Math.sqrt(x);

            return ((sr * sr) == x);
        }
        return false;
    }

    public static void main(String[] args)
```

```

    {
        int x = 84;

        if (isPerfectSquare(x))
            System.out.print("True");
        else
            System.out.print("False");
    }
}

```

## Code in Python

```

Matrix = [[0, 0, 0, 1],
          [0, 1, 1, 1],
          [1, 1, 1, 1],
          [0, 0, 0, 0]]

```

```

max_count, row = 0, -1

```

```

for i in range(4):

```

```

    count = 0
    for j in range(4):
        if Matrix[i][j] == 1:
            count += 1

```

```

    if count > max_count:
        max_count = count
        row = i

```

```

print("Row with maximum 1's is :", row+1)

```

## Q86: Friendly pair

**Write a program to find Friendly pair**

## Code in C

```

#include <stdio.h>

```

```

int getDivisorsSum(int num){

```

```

    int sum = 0;

```

```

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

int main ()
{
    int num1 = 6, num2 = 28;

    int sum1 = getDivisorsSum(num1);
    int sum2 = getDivisorsSum(num2);

    if(sum1/num1 == sum2/num2)
        printf("(%d, %d) are friendly pairs", num1, num2);
    else
        printf("(%d, %d) are not friendly pairs", num1, num2);

}

```

## Code in C++

```

#include <iostream>
using namespace std;

int getDivisorsSum(int num){

    int sum = 0;

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

int main ()
{
    int num1 = 30, num2 = 140;

    int sum1 = getDivisorsSum(num1);
    int sum2 = getDivisorsSum(num2);

```



```

if(sum1/num1 == sum2/num2)
    cout << num1 << " & " << num2 << " are friendly pairs";
else
    cout << num1 << " & " << num2 << " are not friendly pairs";

}

```

## Code in Java

```

public class Main
{
    public static void main (String[]args)
    {

        int num1 = 30, num2 = 140;

        int sum1 = getDivisorsSum (num1);
        int sum2 = getDivisorsSum (num2);

        if (sum1 / num1 == sum2 / num2)
            System.out.println (num1 + " & " + num2 + " are friendly pairs");
        else
            System.out.println (num1 + " & " + num2 + " are not friendly pairs");
    }

    static int getDivisorsSum (int num)
    {

        int sum = 0;

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

```

## Code in Python

```
def printDivisors(n, factors) :
```

```
    i = 1
```

```
    while i <= n :
```

```
        if (n % i==0) :
```

```
            factors.append(i)
```

```
            i = i + 1
```

```
    return sum(factors) - n
```

```
if __name__ == "__main__":
```

```
    number1, number2 = 6, 28
```

```
    if int(printDivisors(number1, [])/number1) == int(printDivisors(number2, [])/number2):
```

```
        print("Friendly pair")
```

```
    else:
```

```
        print("Not a Friendly Pair")
```

## Q87: Print elements in sorted order using row-column wise sorted matrix

Write a program to print elements in sorted order using row-column wise sorted matrix

## Code in C

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

```
int main(){
```

```
    int n=4, m=4;
```

```
    int mat[4][4]= { { 1, 20, 43, 14 },
```

```
                    { 50, 69, 17, 81 },
```

```
                    { 99, 10, 11, 22 },
```

```
                    { 13, 54, 95, 16 } };
```

```
    int arr[16], x=0;
```

```
    for(int i=0; i< n; i++){
```

```
        for(int j=0; j< m; j++){
```

```
            arr[x++]=mat[i][j];
```

```
        }
```

```
    }
```

```

int size = n*m;

for(int i=0; i< size; i++){
    for(int j=i+1; j< size; j++){
        if(arr[i]>arr[j]){
            int temp = arr[i];
            arr[i] = arr[j];
            arr[j] = temp;
        }
    }
}
for(int i=0; i< size; i++)
    printf("%d ",arr[i]);
}

```

## Code in C++

```

#include<bits/stdc++.h>
using namespace std;

int main(){

    int n=4, m=4;
    int mat[n][m]= { { 1, 20, 43, 14 }, { 50, 69, 17, 81 }, { 99, 10, 11, 22 }, { 13, 54, 95, 16 } };

    int arr[n*m], x=0;

    for(int i=0; i<n; i++){
        for(int j=0; j<m; j++){
            arr[x++]=mat[i][j];
        }
    }

    int size = n*m;
    sort(arr, arr+size);

    for(int i=0; i<size; i++)
        cout<<arr[i]<<" ";
}

```

## Code in Java

```

import java.util.Arrays;

class Main{

```

```

public static void main(String args[])
{
    int mat[][] = {{10, 20, 30, 40},
                   {15, 25, 35, 45},
                   {27, 29, 37, 48},
                   {32, 33, 39, 50}};

    int n=4, m=4;

    int[] arr = new int[n*m];
    int x=0;

    for(int i=0; i<n; i++){
        for(int j=0; j<m; j++){
            arr[x++]=mat[i][j];
        }
    }

    int size = n*m;
    Arrays.sort(arr);

    for(int i=0; i<size; i++)
        System.out.print(arr[i] + " ");
}
}

```

## Code in Python

```

Matrix = [[1, 20, 43, 14],
          [50, 69, 17, 81],
          [99, 10, 11, 22],
          [13, 54, 95, 16]]

arr = []
x, n, m = 0, 4, 4

for i in range(n):
    for j in range(m):
        arr.append(Matrix[i][j])

size = n*m
arr.sort()

for i in range(size):

```

```
print(arr[i], end=" ")
```

## Q88: Find a specific pair in matrix

Write a program to find a specific pair in matrix

### Code in C

```
#include <stdio.h>
#include <limits.h>
#define N 5

int findMaxValue(int mat[][N])
{
    int maxValue = INT_MIN;

    for (int a = 0; a < N - 1; a++)
        for (int b = 0; b < N - 1; b++)
            for (int d = a + 1; d < N; d++)
                for (int e = b + 1; e < N; e++)
                    if (maxValue < (mat[d][e] - mat[a][b]))
                        maxValue = mat[d][e] - mat[a][b];

    return maxValue;
}

int main()
{
    int mat[N][N] = {
        { 1, 2, -1, -4, -20 },
        { -8, -3, 4, 2, 1 },
        { 3, 8, 6, 1, 3 },
        { -4, -1, 1, 7, -6 },
        { 0, -4, 10, -5, 1 }
    };

    printf("Maximum Value is %d", findMaxValue(mat));

    return 0;
}
```

### Code in C++

```
#include <iostream>
#include <climits>
```



```

        if (maxValue < (mat[d][e] - mat[a][b]))
            maxValue = mat[d][e] - mat[a][b];

    return maxValue;
}

public static void main (String[] args)
{
    int N = 5;

    int mat[][] = {
        { 1, 2, -1, -4, -20 },
        { -8, -3, 4, 2, 1 },
        { 3, 8, 6, 1, 3 },
        { -4, -1, 1, 7, -6 },
        { 0, -4, 10, -5, 1 }
    };

    System.out.print("Maximum Value is " + findMaxValue(N,mat));
}
}

```

## Code in Python

N = 5

```

def findMaxValue(mat):
    maxValue = 0

    for a in range(N - 1):
        for b in range(N - 1):
            for d in range(a + 1, N):
                for e in range(b + 1, N):
                    if maxValue < (mat[d][e] - mat[a][b]):
                        maxValue = mat[d][e] - mat[a][b]
    return maxValue

```

```

matrix = [[ 1, 2, -1, -4, -20],
          [-8, -3, 4, 2, 1],
          [ 3, 8, 6, 1, 3],
          [-4, -1, 1, 7, -6],
          [ 0, -4, 10, -5, 1]]

```

```
print("Maximum Value is", findMaxValue(matrix))
```

## Q89: Abundant number

**Write a program to check whether a number is Abundant number**

### Code in C

```
#include <stdio.h>

int main ()
{
    int num = 18, sum = 0;

    for(int i = 1; i < num; i++)
    {
        if(num % i == 0)
            sum = sum + i;
    }
    if(sum > num){
        printf("%d is an Abundant Number\n",num);
        printf("Num: %d\nSum: %d\nAbundance: %d", num, sum, (sum-num));
    } else
        printf("%d is not a Abundant Number",num);
}
```

### Code in C++

```
#include <iostream>
using namespace std;

int main ()
{
    int n = 12, sum = 0;

    for(int i = 1; i < n; i++) { if(n % i == 0) sum = sum + i; } if(sum > n){
cout << n << " is an Abundant Number\n";
cout << "The Abundance is: " << (sum-n);
    } else
cout << n << " is not an Abundant Number\n";
}
```



## Code in Java

```
public class Main
{
    public static void main (String[]args)
    {

        int n = 12, sum = 0;

        for (int i = 1; i < n; i++) { if (n % i == 0) sum = sum + i; } if (sum > n)
        {
            System.out.println (n + " is an Abundant Number");
            System.out.println ("The Abundance is: " + (sum - n));
        }
        else
            System.out.println (n + " is not an Abundant Number");
    }
}
```

## Code in Python

```
n = 12

sum=1 # 1 can divide any number

for i in range(2,n):
    if(n%i==0):    #if number is divisible by i add the number
        sum=sum+i

if(sum>n):
    print(n,'is Abundant Number')

else:
    print(n,'is not Abundant Number')
```

## Q90: Rotate matrix by 90 degrees

Write a program to rotate matrix by 90 degrees

## Code in C

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

```

void swap(int *x, int *y){
    int temp = *x;
    *x = *y;
    *y= temp;
}
int main(){

    int n=4;
    int mat[4][4]= { { 1, 2, 3, 4 },{ 5, 6, 7, 8 },{ 9, 10, 11, 12 },{ 13, 14, 15, 16 } };

    //Tranposing the matrix
    for(int i=0; i<n; i++){
        for(int j=i+1; j<n; j++){
            swap(&mat[i][j], &mat[j][i]);
        }

        //Reversing each row of the matrix
        for(int i=0; i<n; i++){
            for(int j=0; j<n/2; j++){
                swap(&mat[i][j], &mat[i][n-j-1]);
            }
        }

        //Print the matrix
        printf("Rotated Matrix :\n");
        for(int i=0; i<n; i++){
            for(int j=0; j<n; j++){
                printf("%d ",mat[i][j]);
            }
            printf("\n");
        }
    }
}

```

## Code in C++

```

#include<bits/stdc++.h>
using namespace std;
int main(){

    int n=4;
    int mat[n][n]= { { 1, 2, 3, 4 },{ 5, 6, 7, 8 },{ 9, 10, 11, 12 },{ 13, 14, 15, 16 } };

    //Tranposing the matrix
    for(int i=0; i<n; i++){

```

```

        for(int j=i+1; j<n; j++)
            swap(mat[i][j], mat[j][i]);
    }

    //Reversing each row of the matrix
    for(int i=0; i<n; i++){
        for(int j=0; j<n/2; j++){
            swap(mat[i][j], mat[i][n-j-1]);
        }
    }

    //Print the matrix
    cout<<"Rotated Matrix :\n";
    for(int i=0; i<n; i++){
        for(int j=0; j<n; j++){
            cout<<mat[i][j]<<" ";
        }
        cout<<endl;
    }
}

```

## Code in Java

```

import java.util.*;

class Main
{

    static void reverseRows (int mat[][])
    {
        int n = mat.length;
        for (int i = 0; i < mat.length; i++){
            for (int j = 0; j < mat.length/ 2; j++){
                int temp = mat[i][j];
                mat[i][j] = mat[i][n - j - 1];
                mat[i][n - j - 1] = temp;
            }
        }
    }

    static void transpose (int arr[][])
    {
        for (int i = 0; i < arr.length; i++)
    }

```

```

        for (int j = i; j < arr[0].length; j++){
            int temp = arr[j][i];
            arr[j][i] = arr[i][j];
            arr[i][j] = temp;
        }
    }

static void printMatrix (int arr[][]){
    for (int i = 0; i < arr.length; i++){
        for (int j = 0; j < arr[0].length; j++){
            System.out.print (arr[i][j] + " ");
            System.out.println ("");
        }
    }

static void rotate90 (int arr[][])
{
    transpose (arr);
    reverseRows (arr);
}

public static void main (String[] args)
{
    int arr[][] = { {1, 2, 3, 4},
                    {5, 6, 7, 8},
                    {9, 10, 11, 12},
                    {13, 14, 15, 16}
    };

    rotate90 (arr);
    printMatrix (arr);
}

```

## Code in Python

n = 4

```

mat = [[1, 2, 3, 4],
       [5, 6, 7, 8],
       [9, 10, 11, 12],
       [13, 14, 15, 16]]

```

# Transposing the matrix

```

for i in range(n):
    for j in range(i + 1, n):
        mat[i][j], mat[j][i] = mat[j][i], mat[i][j]

# Reversing each row of the matrix
for i in range(n):
    for j in range(n // 2):
        mat[i][j], mat[i][n - j - 1] = mat[i][n - j - 1], mat[i][j]

# Print the matrix
print("Rotated Matrix :")
for i in range(n):
    for j in range(n):
        if j == n - 1:
            print(mat[i][j])
        else:
            print(mat[i][j], end=" ")

```

## Q91: Harshad number

Write a program to check Harshad number

### Code in C

```

#include <stdio.h>

int checkHarshad(int num){

    int sum = 0;
    int temp = num;

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

    // will return 1 if num is divisible by sum, else 0
    return num % sum == 0;
}

int main ()
{
    int num = 153;

```

```

    if(checkHarshad(num))
        printf("%d is Harshad's Number", num);
    else
        printf("%d is not Harshad's Number", num);

    return 0;
}

```

## Code in C++

```

#include <iostream>
using namespace std;

int checkHarshad(int num){

    int sum = 0;
    int temp = num;

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

    // will return 1 if num is divisible by sum, else 0
    return num % sum == 0;
}

int main ()
{
    int n = 153;

    if(checkHarshad(n))
        cout << n << " is a Harshad's number";
    else
        cout << n << " is not a Harshad's number";

    return 0;
}

```

## Code in Java

```

public class Main {

```

```

public static void main(String[] args)
{
    //make a copy of original number
    int n = 47;
    //declare a variable to store sum of digits
    int result = 0;
    //perform logic for calculating sum of digits of a number
    while(n != 0)
    {
        int pick_last = n % 10;
        result = result + pick_last;
        n = n / 10;
    }
    /*use condition to check whether the number entered by
    user is completely divisible by its sum of digits or not*/
    if(n % result == 0)
        System.out.println("Harshad Number");
    else
        System.out.println("Not a Harshad Number");
}
}

```

## Code in Python

```

n = 21
p=n
l=[]
sum1=0
while(n>0):
    x=n%10
    l.append(x)
    n=n//10
sum1=sum(l)
if(p%sum1==0):
    print("Harshad number")
else:
    print("Not harshad number")

```

## Q92: Kth smallest element in a row-column wise sorted matrix

Write a program to Kth smallest element in a row-column wise sorted matrix

### Code in C

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

struct HeapNode {
    int val;
    int r;
    int c;
};

void minHeapify(struct HeapNode harr[], int i, int heap_size) {
    int l = i * 2 + 1;
    int r = i * 2 + 2;

    if (l < heap_size && r < heap_size && harr[l].val < harr[i].val && harr[r].val < harr[i].val) {
        struct HeapNode temp = harr[r];
        harr[r] = harr[i];
        harr[i] = harr[l];
        harr[l] = temp;
        minHeapify(harr, l, heap_size);
        minHeapify(harr, r, heap_size);
    }

    if (l < heap_size && harr[l].val < harr[i].val) {
        struct HeapNode temp = harr[i];
        harr[i] = harr[l];
        harr[l] = temp;
        minHeapify(harr, l, heap_size);
    }
}

int kthSmallest(int mat[4][4], int n, int k) {
    if (k < 0 || k >= n * n)
        return INT_MAX;

    struct HeapNode harr[n];
    for (int i = 0; i < n; i++) {
        harr[i].val = mat[0][i];
```



```

        harr[i].r = 0;
        harr[i].c = i;
    }

    struct HeapNode hr;
    for (int i = 0; i < k; i++) {
        hr = harr[0];

        int nextval = (hr.r < (n - 1)) ? mat[hr.r + 1][hr.c] : INT_MAX;

        harr[0].val = nextval;
        harr[0].r = hr.r + 1;
        harr[0].c = hr.c;

        minHeapify(harr, 0, n);
    }

    return hr.val;
}

int main() {
    int mat[4][4] = {
        { 10, 20, 30, 40 },
        { 15, 25, 35, 45 },
        { 25, 29, 37, 48 },
        { 32, 33, 39, 50 },
    };
    printf("6th smallest element is %d\n", kthSmallest(mat, 4, 6));
    return 0;
}

```

## Code in C++

```

#include <bits/stdc++.h>
using namespace std;

struct HeapNode {
    int val;
    int r;
    int c;
};

void minHeapify(HeapNode harr[], int i, int heap_size)
{
    int l = i * 2 + 1;

```

```

int r = i * 2 + 2;
if(l < heap_size && r < heap_size && harr[l].val < harr[i].val && harr[r].val < harr[i].val){
    HeapNode temp=harr[r];
    harr[r]=harr[i];
    harr[i]=harr[l];
    harr[l]=temp;
    minHeapify(harr ,l,heap_size);
    minHeapify(harr ,r,heap_size);
}
if (l < heap_size && harr[l].val < harr[i].val){
    HeapNode temp=harr[i];
    harr[i]=harr[l];
    harr[l]=temp;
    minHeapify(harr ,l,heap_size);
}
}

```

```

int kthSmallest(int mat[4][4], int n, int k)
{
    if (k < 0 || k >= n * n)
        return INT_MAX;

    HeapNode harr[n];
    for (int i = 0; i < n; i++)
        harr[i] = { mat[0][i], 0, i };

    HeapNode hr;
    for (int i = 0; i < k; i++) {

        hr = harr[0];

        int nextval = (hr.r < (n - 1)) ? mat[hr.r + 1][hr.c]: INT_MAX;

        harr[0] = { nextval, (hr.r) + 1, hr.c };

        minHeapify(harr, 0, n);
    }

    return hr.val;
}

```

```

int main()
{
    int mat[4][4] = {

```

```

        { 10, 20, 30, 40 },
        { 15, 25, 35, 45 },
        { 25, 29, 37, 48 },
        { 32, 33, 39, 50 },
    };
    cout << "6th smallest element is "<< kthSmallest(mat, 4, 6);
    return 0;
}

```

## Code in Java

```

class Main{
    static class HeapNode
    {

        int val;
        int r;
        int c;

        HeapNode(int val, int r, int c)
        {
            this.val = val;
            this.c = c;
            this.r = r;
        }
    }

    static void minHeapify(HeapNode harr[], int i, int heap_size)
    {
        int l = 2 * i + 1;
        int r = 2 * i + 2;
        int min = i;

        if(l < heap_size && r<heap_size && harr[l].val < harr[i].val && harr[r].val < harr[i].val){
            HeapNode temp=harr[r];
            harr[r]=harr[i];
            harr[i]=harr[l];
            harr[l]=temp;
            minHeapify(harr ,l,heap_size);
            minHeapify(harr ,r,heap_size);
        }
        if (l < heap_size && harr[l].val < harr[i].val){
            HeapNode temp=harr[i];
            harr[i]=harr[l];
            harr[l]=temp;
        }
    }
}

```

```

        minHeapify(harr ,l,heap_size);
    }
}

public static int kthSmallest(int[][] mat,int n, int k)
{
    if (k < 0 && k >= n * n)
        return Integer.MAX_VALUE;

    HeapNode harr[] = new HeapNode[n];

    for(int i = 0; i < n; i++)
    {
        harr[i] = new HeapNode(mat[0][i], 0, i);
    }

    HeapNode hr = new HeapNode(0, 0, 0);

    for(int i = 1; i <= k; i++)
    {
        hr = harr[0];

        int nextVal = hr.r < n - 1 ? mat[hr.r + 1][hr.c] : Integer.MAX_VALUE;

        harr[0] = new HeapNode(nextVal, hr.r + 1, hr.c);

        minHeapify(harr, 0, n);
    }

    return hr.val;
}

// Driver code
public static void main(String args[])
{
    int mat[][] = { { 10, 20, 30, 40 },
                    { 15, 25, 35, 45 },
                    { 25, 29, 37, 48 },
                    { 32, 33, 39, 50 } };

    int res = kthSmallest(mat, 4, 6);

```

```

        System.out.print("6th smallest element is "+ res);
    }
}

```

## Code in Python

```

import heapq

def kth_smallest(mat, n, k):
    min_heap = [(mat[0][i], 0, i) for i in range(n)]
    heapq.heapify(min_heap)

    for _ in range(k):
        val, r, c = heapq.heappop(min_heap)
        if r < n - 1:
            heapq.heappush(min_heap, (mat[r + 1][c], r + 1, c))

    return val

if __name__ == "__main__":
    mat = [
        [10, 20, 30, 40],
        [15, 25, 35, 45],
        [25, 29, 37, 48],
        [32, 33, 39, 50],
    ]
    print("6th smallest element is", kth_smallest(mat, 4, 6))

```

## Q93: Automorphic number

Write a program to check Automorphic number

## Code in C

```

#include <stdio.h>

int checkAutomorphic(int num){

    int square = num * num;

    while(num != 0)
    {
        // means not automorphic number
        if(num % 10 != square % 10){

```

```

        return 0;
    }

    // reduce down numbers
    num /= 10;
    square /= 10;
}
// if reaches here means automorphic number
return 1;
}

int main ()
{
    int num = 376, square = num * num ;

    if(checkAutomorphic(num))
        printf("Num : %d, Square: %d - Automorphic Number",num, square);
    else
        printf("Num : %d, Square: %d - Not Automorphic Number",num, square);

}

```

## Code in C++

```

#include <iostream>
using namespace std;

int isAutomorphic(int n){

    int square = n * n;

    while(n != 0)
    {
        // means not automorphic number
        if(n % 10 != square % 10){
            return 0;
        }

        // reduce down numbers
        n /= 10;
        square /= 10;
    }
    // if reaches here means automorphic number
}

```

```

    return 1;
}

int main ()
{
    int n = 376, sq = n * n ;

    if(isAutomorphic(n))
        cout << "Num: "<< n << ", Square: " << sq << " - is Automorphic";
    else
        cout << "Num: "<< n << ", Square: " << sq << " - is not Automorphic";

}

```

## Code in Java

```

public class Main
{
    public static void main(String[] args) {

        int n = 376, sq = n * n ;
        if(isAutomorphic(n) == 1)
            System.out.println("Num: "+ n + ", Square: " + sq + " - is Automorphic");
        else
            System.out.println("Num: "+ n + ", Square: " + sq + " - is not Automorphic");

    }

    static int isAutomorphic(int n){
        int square = n * n;
        while(n != 0)
        {
            // means not automorphic number
            if(n % 10 != square % 10){
                return 0;
            }
            // reduce down numbers
            n /= 10;
            square /= 10;
        }
        // if reaches here means automorphic number
        return 1;
    }
}

```

```
}  
}
```

## Code in Python

```
number = 376  
square = pow(number, 2)  
mod = pow(10, len(str(number)))  
  
# 141376 % 1000  
if square % mod == number:  
    print("It's an Automorphic Number")  
else:  
    print("It's not an Automorphic Number")
```

## Q94: Common elements in all rows of a given matrix

Write a program to find common elements in all rows of a given matrix

### Code in C

```
#include <stdio.h>  
#define M 4  
#define N 5  
  
void printCommonElements(int mat[M][N])  
{  
  
    for (int j = 0; j < N; j++)  
    {  
        int x = mat[0][j], count = 0;  
  
        for (int i = 1; i < M; i++){  
            int flag = 0;  
            for(int j = 0; j < N; j++){  
                if(x==mat[i][j]){  
                    flag = 1;  
                    mat[i][j] = -1;  
                    break;  
                }  
            }  
            if(flag==1){
```



```

        count++;
    }
}

if (count==M-1)
    printf("%d ", x);

}

}

int main()
{
    int mat[M][N] =
    {
        {10, 20, 10, 40, 80},
        {30, 70, 80, 50, 10},
        {80, 70, 70, 30, 10},
        {80, 10, 20, 70, 90},
    };

    printCommonElements(mat);

    return 0;
}

```

## Code in C++

```

#include <bits/stdc++.h>
using namespace std;
#define N 5

int findMaxValue(int mat[][N])
{
    int maxVal = INT_MIN;

    for (int a = 0; a < N - 1; a++)
        for (int b = 0; b < N - 1; b++)
            for (int d = a + 1; d < N; d++)
                for (int e = b + 1; e < N; e++)
                    if (maxVal < (mat[d][e] - mat[a][b]))
                        maxVal = mat[d][e] - mat[a][b];

    return maxVal;
}

```

```

int main()
{
int mat[N][N] = {
    { 1, 2, -1, -4, -20 },
    { -8, -3, 4, 2, 1 },
    { 3, 8, 6, 1, 3 },
    { -4, -1, 1, 7, -6 },
    { 0, -4, 10, -5, 1 }
};
cout << "Maximum Value is " << findMaxValue(mat);

return 0;
}

```

## Code in Java

```

class Main
{

public static void main(String args[])
{
    int mat[][] = {{10, 20, 30, 40},
                    {15, 25, 35, 30},
                    {27, 30, 37, 48},
                    {32, 33, 39, 30}};

    int N=4, M=4;

    for (int j = 0; j < N; j++){
        int x = mat[0][j], count = 0;

        for (int i = 1; i < M; i++){
            int flag = 0;
            for(int k = 0; k < N; k++){
                if(x==mat[i][k]){
                    flag = 1;
                    mat[i][k] = -1;
                    break;
                }
            }
            if(flag==1){
                count++;
            }
        }
    }
}
}

```

```

        if (count==M-1)
            System.out.print(x);

    }
}
}

```

## Code in Python

```

def find(mat, N, M):
    for j in range(0, N):
        x, count = mat[0][j], 0

        for i in range(1, M):
            flag = 0
            for k in range(0, N):
                if x == mat[i][k]:
                    flag = 1
                    mat[i][k] = -1
                    break
            if flag == 1:
                count += 1
        if count == M - 1:
            print(x)

```

```

N, M = 4, 4
mat = [[10, 20, 30, 40],
        [15, 25, 35, 30],
        [27, 30, 37, 48],
        [32, 33, 39, 30]]
find(mat, N, M)

```

## Q95: Strong number

Write a program to check whether a Strong number

## Code in C

```

#include <stdio.h>

// function to calculate factorial
int getFactorial(int n){

```

```

int fact = 1;

for(int i = 1; i <= n; i++)
    fact = fact * i;

return fact;
}

int checkStrong(int num){

    int digit, sum = 0;
    int temp = num;

    // calculate 1! + 4! + 5!
    while(temp!=0){
        digit = temp % 10;

        sum = sum + getFactorial(digit);
        temp /= 10;
    }

    // returns 1 if both equal else 0
    return sum == num;

}

int main ()
{
    int num = 145;

    if(checkStrong(num))
        printf("%d is Strong Number", num);
    else
        printf("%d is Not Strong Number", num);

}

```

## Code in C++

```

#include <iostream>
using namespace std;

// function to calculate factorial
int facto(int n){
    int fact = 1;

```

```

    for(int i = 1; i <= n; i++)
        fact = fact * i;

    return fact;
}

int detectStrong(int num){

    int digit, sum = 0;
    int temp = num;

    // calculate 1! + 4! + 5!
    while(temp!=0){
        digit = temp % 10;

        sum = sum + facto(digit);
        temp /= 10;
    }

    // returns 1 if both equal else 0
    return sum == num;

}

int main ()
{
    int num = 145;

    if(detectStrong(num))
        cout << num << " is Strong Number";
    else
        cout << num << " is Not Strong Number";

}

```

## Code in Java

```

public class Main
{
    public static void main (String[]args)
    {

        int num = 145;

```

```

    if (detectStrong (num))
        System.out.println (num + " is Strong Number");
    else
        System.out.println (num + " is not a Strong Number");
}

// function to calculate factorial
static int facto (int n)
{
    int fact = 1;

    for (int i = 1; i <= n; i++)
        fact = fact * i;

    return fact;
}

static boolean detectStrong (int num)
{
    int digit, sum = 0;
    int temp = num;
    boolean flag = false;

    // calculate 1! + 4! + 5!
    while (temp != 0)
    {
        digit = temp % 10;

        sum = sum + facto (digit);
        temp /= 10;
    }

    // returns 1 if both equal else 0
    if (sum == num)
        flag = true;
    else
        flag = false;

    return flag;
}
}

```

## Code in Python

```
#Using Iteration
n =145
#save the number in another variable
temp=n
sum=0
f=[0]*10
f[0]=1
f[1]=1
for i in range(2,10): #precomputing the factorial value from 0 to 9 and store in the array.
    f[i]=f[i-1]*i

#Implementation
while(temp):
    r=temp%10 #r will have the value of the unit digit
    temp=temp//10
    sum+=f[r] #adding all the factorial

if(sum==n):
    print("Yes", n ,"is strong number")

else:
    print("No" , n, "is not a strong number")
```

## Q96: Find all the N bit binary numbers having more than or equal 1's than 0's

Write a program to find all the N bit binary numbers having more than or equal 1's than 0's

## Code in C

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

// Recursive function to print the required numbers
void printRec(char number[], int extraOnes, int remainingPlaces) {
    if (remainingPlaces == 0) {
        printf("%s ", number);
    }
```

```

        return;
    }

    int length = strlen(number);
    if (length == 0 || number[length - 1] == '0') {
        number[length] = '1';
        number[length + 1] = '\0';
        printRec(number, extraOnes + 1, remainingPlaces - 1);
        number[length] = '\0';
    }

    if (extraOnes > 0) {
        number[length] = '0';
        number[length + 1] = '\0';
        printRec(number, extraOnes - 1, remainingPlaces - 1);
        number[length] = '\0';
    }
}

// Driver code
int main() {
    int n = 4;
    char str[5] = ""; // 4 digits and a null terminator

    printRec(str, 0, n);

    return 0;
}

```

## Code in C++

```

#include <bits/stdc++.h>
using namespace std;

//Recursive function to print the required numbers
void printRec(string number, int extraOnes, int remainingPlaces)
{
    if (remainingPlaces == 0) {
        cout << number << " "; return;
    }
    printRec(number + "1", extraOnes + 1, remainingPlaces - 1);

    if (extraOnes > 0)
        printRec(number + "0", extraOnes - 1, remainingPlaces - 1);
}

```



```
// Driver code
int main()
{
    int n = 4;

    string str = "";
    printRec(str, 0, n);

    return 0;
}
```

## Code in Java

```
import java.io.*;

class Main {
    static void printRec(String number, int extraOnes, int remainingPlaces)
    {
        if (0 == remainingPlaces) {
            System.out.print(number + " ");
            return;
        }

        printRec(number + "1", extraOnes + 1, remainingPlaces - 1);

        if (0 < extraOnes)
            printRec(number + "0", extraOnes - 1, remainingPlaces - 1);
    }

    static void printNums(int n)
    {
        String str = "";
        printRec(str, 0, n);
    }

    public static void main(String[] args)
    {
        int n = 4;

        printNums(n);
    }
}
```

## Code in Python

```
def printRec(number, extraOnes, remainingPlaces):
    if 0 == remainingPlaces:
        print(number, end=" ")
        return

    printRec(number + "1", extraOnes + 1, remainingPlaces - 1)

    if 0 < extraOnes:
        printRec(number + "0", extraOnes - 1, remainingPlaces - 1)

def printNums(n):
    str = ""
    printRec(str, 0, n)

n = 4
printNums(n)
```

## Q97: Fibonacci Series upto *nth* term

Write a program to find fibonacci Series upto *nth* term

## Code in C

```
#include<stdio.h>

int main()
{
    int n = 10;
    int a = 0, b = 1;

    // printing the 0th and 1st term
    printf("%d, %d",a,b);

    int nextTerm;

    // printing the rest of the terms here
    for(int i = 2; i < n; i++){
        nextTerm = a + b;
        a = b;
        b = nextTerm;
```

```

        printf("%d, ",nextTerm);
    }

    return 0;
}

```

## Code in C++

```

#include <iostream>
using namespace std;

int main()
{
    int num = 15;
    int a = 0, b = 1;

    // Here we are printing 0th and 1st terms
    cout << a << ", " << b << ", ";

    int nextTerm;

    // printing the rest of the terms here
    for(int i = 2; i < num; i++){
        nextTerm = a + b;
        a = b;
        b = nextTerm;

        cout << nextTerm << ", ";
    }

    return 0;
}

```

## Code in Java

```

public class Main
{
    public static void main (String[]args)
    {

        int num = 15;
        int a = 0, b = 1;
    }
}

```

```

// Here we are printing 0th and 1st terms
System.out.print (a + " , " + b + " , ");

int nextTerm;

// printing the rest of the terms here
for (int i = 2; i < num; i++)
{
    nextTerm = a + b;
    a = b;
    b = nextTerm;
    System.out.print (nextTerm + " , ");
}

}
}

```

## Code in Python

```

num = 10
n1, n2 = 0, 1
print("Fibonacci Series:", n1, n2, end=" ")
for i in range(2, num):
    n3 = n1 + n2
    n1 = n2
    n2 = n3
    print(n3, end=" ")

print()

```

## Q98: Given a set of positive integers, find all its subsets

Write a program to give a set of positive integers, find all its subsets

## Code in C

```

#include <stdio.h>
#include <stdlib.h>

// Define a structure for a dynamic array

```

```

typedef struct {
    int* arr;
    int size;
    int capacity;
} DynamicArray;

// Initialize a dynamic array
void initDynamicArray(DynamicArray* dynArr) {
    dynArr->capacity = 1;
    dynArr->size = 0;
    dynArr->arr = (int*)malloc(sizeof(int) * dynArr->capacity);
}

// Append an element to the dynamic array
void appendDynamicArray(DynamicArray* dynArr, int value) {
    if (dynArr->size == dynArr->capacity) {
        dynArr->capacity *= 2;
        dynArr->arr = (int*)realloc(dynArr->arr, sizeof(int) * dynArr->capacity);
    }
    dynArr->arr[dynArr->size] = value;
    dynArr->size++;
}

// Recursive function to generate subsets
void subsetsUtil(int* A, int n, DynamicArray* subset, DynamicArray* res, int index) {
    for (int i = index; i < n; i++) {
        appendDynamicArray(subset, A[i]);
        for (int j = 0; j < subset->size; j++) {
            appendDynamicArray(res, subset->arr[j]);
        }
        subsetsUtil(A, n, subset, res, i + 1);
        subset->size--;
    }
}

// Generate all subsets of an array
DynamicArray subsets(int* A, int n) {
    DynamicArray subset, res;
    initDynamicArray(&subset);
    initDynamicArray(&res);

    subsetsUtil(A, n, &subset, &res, 0);

    return res;
}

```

```

}

int main() {
    int array[] = {1, 2, 3};
    int n = sizeof(array) / sizeof(array[0]);

    DynamicArray res = subsets(array, n);

    for (int i = 0; i < res.size; i++) {
        printf("{ ");
        for (int j = i; j < res.size; j++) {
            printf("%d ", res.arr[j]);
        }
        printf("}\n");
    }

    free(res.arr);

    return 0;
}

```

## Code in C++

```

#include <bits/stdc++.h>
using namespace std;

void subsetsUtil(vector<int>& A, vector<vector<int> >& res, vector<int>& subset, int index)
{
    res.push_back(subset);
    for (int i = index; i < A.size(); i++) {

        subset.push_back(A[i]);
        subsetsUtil(A, res, subset, i + 1);
        subset.pop_back();
    }

    return;
}

vector<vector<int> > subsets(vector<int>& A)
{
    vector<int> subset;
    vector<vector<int> > res;

    int index = 0;

```

```

    subsetsUtil(A, res, subset, index);

    return res;
}

int main()
{
    vector<int> array = { 1, 2, 3 };

    vector<vector<int> > res = subsets(array);

    for (int i = 0; i < res.size(); i++) {
        for (int j = 0; j < res[i].size(); j++)
            cout << res[i][j] << " ";
        cout << endl;
    }

    return 0;
}

```

## Code in Java

```

import java.io.*;
import java.util.*;
class Main {
    public static void
    findSubsets(List<List<Integer>> subset, ArrayList<Integer> nums, ArrayList<Integer> output,
int index)
    {
        if (index == nums.size()) {
            subset.add(output);
            return;
        }

        findSubsets(subset, nums, new ArrayList<>(output), index + 1);

        output.add(nums.get(index));
        findSubsets(subset, nums, new ArrayList<>(output), index + 1);
    }

    public static void main(String[] args) {

        List<List<Integer>> subset = new ArrayList<>();

        ArrayList input = new ArrayList<>();
    }
}

```

```

input.add(1);
input.add(2);
input.add(3);

```

```

findSubsets(subset, input, new ArrayList<>(), 0);

```

```

Collections.sort(subset, (o1, o2) -> {
    int n = Math.min(o1.size(), o2.size());
    for (int i = 0; i < n; i++) {
        if (o1.get(i) == o2.get(i)){
            continue;
        }else{
            return o1.get(i) - o2.get(i);
        }
    }
    return 1;
});

```

```

for(int i = 0; i < subset.size(); i++){
    for(int j = 0; j < subset.get(i).size(); j++){
        System.out.print(subset.get(i).get(j) + " ");
    }
    System.out.println();
}
}
}

```

## Code in Python

```

def subsetsUtil(A, subset=[], index=0):

```

```

    print(*subset)

```

```

    for i in range(index, len(A)):
        subset.append(A[i])

```

```

        subsetsUtil(A, subset, i + 1)

```

```

    subset.pop(-1)
    return

```

```

array = [1, 2, 3]

```



subsetsUtil(array)

## Q99: Armstrong number in a given range

Write a program to find Armstrong number in a given range

### Code in C

```
#include<stdio.h>
#include<math.h>
using namespace std;
// number of digits in a number is order
int order(int x)
{
    int len = 0;
    while (x)
    {
        len++;
        x = x/10;
    }
    return len;
}

void armstrong(int low, int high){

    for(int num = low; num <= high; num++){

        int sum = 0, temp, digit, len;
        temp = num;

        // function to get order(length)
        len = order(num);

        // loop to extract digit, find powers & add to sum
        while(temp != 0)
        {
            // extract digit
            digit = temp % 10;

            // add power to sum
            sum = sum + pow(digit,len);
            temp /= 10;
        };
    }
}
```

```

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

// Driver Code
int main ()
{
    int low, high;

    printf("Enter a lower & upper bounds: ");
    scanf("%d %d",&low,&high);

    // get armstrong numbers
    armstrong(low, high);
}

```

## Code in C++

```

#include<bits/stdc++.h>
#include<math.h>
using namespace std;

// number of digits in a number is order
int order(int x)
{
    int len = 0;
    while (x)
    {
        len++;
        x = x/10;
    }
    return len;
}

void armstrong(int low, int high){

    for(int num = low; num <= high; num++){

        int sum = 0, temp, digit, len;
        temp = num;

        // function to get order(length)

```

```

len = order(num);

// loop to extract digit, find cube & add to sum
while(temp != 0)
{
    // extract digit
    digit = temp % 10;

    // add power to sum
    sum = sum + pow(digit,len);
    temp /= 10;
};

if(sum == num)
    cout << num << " ";
}
}

// Driver Code
int main ()
{
    //variables initialization
    int low = 100, high = 400;
    // get armstrong numbers
    armstrong(low, high);

    return 0;
}

```

## Code in Java

```

import java.util.Scanner;

public class LearnCoding2
{
    public static void main(String[] args)
    {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter lower and upper ranges : ");
        int low = sc.nextInt();
        int high = sc.nextInt();

        System.out.println("Armstrong numbers between "+ low + " and " + high + " are : ");
    }
}

```

```

// loop for finding and printing all armstrong numbers between given range
for(int num = low ; num <= high ; num++)
{
    int sum = 0, temp, digit, len;

    len = getOrder(num);
    temp = num;
    // loop to extract digit, find ordered power of digits & add to sum
    while(temp != 0)
    {
        // extract digit
        digit = temp % 10;

        // add power to sum
        sum = sum + (int) Math.pow(digit,len);
        temp /= 10;
    };

    if(sum == num)
        System.out.print(num + " ");
    }
}

private static int getOrder(int num) {
    int len = 0;
    while (num!=0)
    {
        len++;
        num = num/10;
    }
    return len;
}
}

```

## Code in Python

```
low, high = 10, 10000
```

```
for n in range(low, high + 1):
```

```

    # order of number
    order = len(str(n))

```

```

# initialize sum
sum = 0

temp = n
while temp > 0:
    digit = temp % 10
    sum += digit ** order
    temp //= 10

if n == sum:
    print(n, end=" ")

```

## Q100: Given a string s, remove all its adjacent duplicate characters recursively

Write a program to give a string s, remove all its adjacent duplicate characters recursively

### Code in C

```

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

char* removeDuplicates(char* s, char ch) {
    int len = strlen(s);
    if (len <= 1) {
        return s;
    }

    char* result = (char*)malloc(len * sizeof(char));
    int resultIndex = 0;

    int i = 0;
    while (i < len) {
        if (i + 1 < len && s[i] == s[i + 1]) {
            int j = i;
            while (j + 1 < len && s[j] == s[j + 1]) {
                j++;
            }
            char lastChar = i > 0 ? s[i - 1] : ch;

            char* remStr = removeDuplicates(&s[j + 1], lastChar);

```

```

    for (int k = 0; k < i; k++) {
        result[resultIndex] = s[k];
        resultIndex++;
    }

    int k = i;

    while (strlen(remStr) > 0 && k >= 0 && remStr[0] == result[k - 1]) {
        while (strlen(remStr) > 0 && remStr[0] != ch && remStr[0] == result[k - 1]) {
            remStr++;
        }
        resultIndex--;
        k--;
    }

    while (strlen(remStr) > 0) {
        result[resultIndex] = remStr[0];
        resultIndex++;
        remStr++;
    }
    i = j;
} else {
    result[resultIndex] = s[i];
    resultIndex++;
    i++;
}
}
result[resultIndex] = '\0';

return result;
}

int main() {
    char str1[] = "aabbbaacdffd";
    char ch = ' ';
    char* result = removeDuplicates(str1, ch);
    printf("%s\n", result);
    free(result);

    return 0;
}

```

## Code in C++

```
#include <bits/stdc++.h>
using namespace std;

string removeDuplicates(string s, char ch)
{
    // Base condition
    if (s.length() <= 1) {
        return s;
    }

    int i = 0;
    while (i < s.length()) {
        if (i + 1 < s.length() && s[i] == s[i + 1]) {
            int j = i;
            while (j + 1 < s.length() && s[j] == s[j + 1]) {
                j++;
            }
            char lastChar = i > 0 ? s[i - 1] : ch;

            string remStr = removeDuplicates(
                s.substr(j + 1, s.length()), lastChar);

            s = s.substr(0, i);
            int k = s.length(), l = 0;

            while (remStr.length() > 0 && s.length() > 0 && remStr[0] == s[s.length() - 1])
            {
                while (remStr.length() > 0 && remStr[0] != ch && remStr[0] == s[s.length() - 1]) {
                    remStr = remStr.substr(1, remStr.length());
                }
                s = s.substr(0, s.length() - 1);
            }
            s = s + remStr;
            i = j;
        }
        else {
            i++;
        }
    }
    return s;
}
```

```

}

// Driver Code
int main()
{

    string str1 = "aabbbaacdffd";
    cout << removeDuplicates(str1, ' ') << endl;
}

```

## Code in Java

```

public class Main {

    public static String removeDuplicates(String s, char ch) {
        // Base condition
        if (s.length() <= 1) {
            return s;
        }

        StringBuilder result = new StringBuilder();
        int i = 0;
        while (i < s.length()) {
            if (i + 1 < s.length() && s.charAt(i) == s.charAt(i + 1)) {
                int j = i;
                while (j + 1 < s.length() && s.charAt(j) == s.charAt(j + 1)) {
                    j++;
                }
                char lastChar = (i > 0) ? s.charAt(i - 1) : ch;

                String remStr = removeDuplicates(s.substring(j + 1), lastChar);

                result.append(s, 0, i);

                int k = i;
                while (remStr.length() > 0 && k > 0 && remStr.charAt(0) == result.charAt(k - 1)) {
                    while (remStr.length() > 0 && remStr.charAt(0) != ch && remStr.charAt(0) ==
result.charAt(k - 1)) {
                        remStr = remStr.substring(1);
                    }
                    result.deleteCharAt(k - 1);
                    k--;
                }

                result.append(remStr);
            }
            i++;
        }
    }
}

```



```

        i = j;
    } else {
        result.append(s.charAt(i));
        i++;
    }
}
return result.toString();
}

public static void main(String[] args) {
    String str1 = "aabbbaacdffd";
    char ch = ' ';
    String result = removeDuplicates(str1, ch);
    System.out.println(result);
}
}

```

## Code in Python

```

def removeUtil(string, last_removed):

    if len(string) == 0 or len(string) == 1:
        return string

    if string[0] == string[1]:
        last_removed = ord(string[0])

        while len(string) > 1 and string[0] == string[1]:
            string = string[1:]

        string = string[1:]
        return removeUtil(string, last_removed)

    rem_str = removeUtil(string[1:], last_removed)

    if len(rem_str) != 0 and rem_str[0] == string[0]:
        last_removed = ord(string[0])
        return rem_str[1:]

    if len(rem_str) == 0 and last_removed == ord(string[0]):
        return rem_str

    return [string[0]] + rem_str

```

```
def remove(string):  
    last_removed = 0  
    return "".join(removeUtil(list(string), last_removed))
```

```
string1 = "acaabbbceddd"  
print(remove(string1))
```