**Programs I've done -**

1 - Reverse a Number

2 - Palindrome Number

3 - sum of digits

4 - Prime No. program

5 - All Prime NNumbers b/a a & b

**Basic CPP Fundamentals -**

#include<iostream>

using namespace std;

int main()

{

//     cout<<"51 LPA Get Placed\n";//51 LPA Get Placed

//     cout<<5;//51 LPA Get Placed

//     cout<<10+5;//15

//     cout<<20\*5;//100

//     cout<<20/5;//4

// // Data type basics

//     int variable = 50;

//     cout<<variable;//50

//     float variable1 = 50.50;

//     cout<<variable1;//50.50

//     char var = 'A';

//     cout<<var;//A

//     bool var2 = true;

//     cout<<var2;//1 - true is consider as 1 in cpp

    // int a = 10;

    // int b = 100;

    // int c = 1000;

    // cout<<a<<endl;//10

    // cout<<b<<endl;//100

    // cout<<c<<endl;//1000

    // cout<<a+b+c<<endl;//1110

// Taking input in cpp

    // int var3;

    // cout<<"ENter the var3 value"<<endl;

    // cin>>var3;//5000

    // cout<<var3;

// Loops in CPP -

    // print the numbers from 1 to 111

    // for (int i = 1; i <= 111; i++)

    // {

    //     cout<<i<<endl;

    // }

    // int j = 1;

    // while (j<=15)

    // {

    //     cout<<j<<endl;

    //     j++;

    // }

// \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_

// reverse the number

    // int num;

    // cout<<"Enter the no. you want to reverse"<<endl;

    // cin>>num;//372

    // int Reversenum = 0;

    // while (num!=0)

    // {

    //     int last\_digit = num%10;

    //     Reversenum = Reversenum\*10+last\_digit;

    //     num = num/10;

    // }

    // if (Reversenum==num)

    // {

    //     cout<<"Yes..!! Palindrome number"<<endl;

    // }

    // else

    // cout<<"Reversed No. is - "<<Reversenum<<endl;

// \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_

// Palindrome Number - no. having same value when reverse

    // int num;

    // cout<<"Enter the no. you want to check that its Palindrome or not"<<endl;

    // cin>>num;

    // int reversenum = 0;

    // int originalnum = num;

    // while (num!=0)

    // {

    //     int last\_digit = num%10;

    //     reversenum = reversenum\*10+last\_digit;

    //     num = num/10;

    // }

    // if (reversenum == originalnum)

    // {

    //     cout<<"congrats you got a PALINDROME NUMBER"<<endl;

    // }

    // else

    // cout<<"The reverse order of inserted number is " <<reversenum<<"and it's not a palindrome number"<<endl;

// \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_

// 3 - sum of digits

    // int num2;

    // cout<<"Numvber ka sum"<<endl;

    // cin>>num2;

    // int sum=0;

    // while (num2!=0)

    // {

    //      int last\_digit = num2%10;

    //      sum = sum+last\_digit;

    //      num2 /=10;

    // }

    // cout<<sum<<endl;

// \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_

// Prime Number Program

// No. either divide by itself or 1 is prime number - so if no is divisible by 2 to number-1 any number -> Then will not be a prime number

// int num;

// cout<<"Enter the no. you wants to check for Prime or Not"<<endl;

// cin>>num;

// if (num==1)

// {

//     cout<<"No. is Neither Prime Nor Compite"<<endl;

// }

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

// {

//     if (num%i==0)

//     {

//         cout<<"Not a Prime Number"<<endl;

//         return 0;

//     }

// }

// cout<<"prime number"<<endl;

}

// \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_

**Functions in CPP –**

#include <iostream>

using namespace std;

// Functions in C++

// int reverse(int num)

// {

//    int Reversenum = 0;

//    while (num!=0)

//    {

//        int last\_digit = num%10;

//        Reversenum = Reversenum\*10+last\_digit;

//        num = num/10;

//    }

//    return Reversenum;

// }

// int main()

// {

//    // reverse number calling the program for execution

//    int num;

//    cout<<"Enter the no. you want to reverse"<<endl;

//    cin>>num;

//    cout<<reverse(num)<<endl;

//    // now by using the function concept i can do the same for multiple times

//    cout<<reverse(num)<<endl;

//    cout<<reverse(555)<<endl;

//    cout<<reverse(658)<<endl;

//    cout<<reverse(1010)<<endl;

// }

// \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_

//  Number sum -

// int digit\_sum(int num2)

// {

//     int sum = 0;

//     while (num2 != 0)

//     {

//         int last\_digit = num2 % 10;

//         sum = sum + last\_digit;

//         num2 = num2/10;

//     }

//     return sum;

// }

// int main()

// {

//     cout<<digit\_sum(1257)<<endl;

//     cout<<digit\_sum(3458)<<endl;

//     cout<<digit\_sum(555)<<endl;

// }

// \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_

// Prime Number Program

// bool isprime(int num)

// {

//     if (num==1)

// {

//     cout<<"No. is Neither Prime Nor Composite"<<endl;

//     return 0;

// }

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

// {

//     if (num%i==0)

//     {

//         cout<<"Not a Prime Number"<<endl;

//         return 0;

//     }

// }

//     cout<<"prime number"<<endl;

// }

// int main()

// {

//     int num;

//     cout<<"Enter the no. you wants to check for Prime or Not"<<endl;

//     cin>>num;

//     isprime(num);

//     cout<<isprime(num)<<endl;

// }

// \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_

// bool isprime(int num)

// {

//     if (num==1)//it based on the qun as per given treat 1 ptime or not. here treating 1 as prime

// {

//     return true;

// }

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

// {

//     if (num%i==0)

//     {

//         return false;

//     }

// }

//     return true;

// }

// int main()

// {

//     int num;

//     cout<<"Enter the no. you wants to check for Prime or Not"<<endl;

//     cin>>num;

//     isprime(num);

//     cout<<isprime(num)<<endl;

// }

// \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_

// Print all Prime Numbers between a and b

bool isprime(int num)

{

    if (num == 1)

    {

        return true;

    }

    for (int i = 2; i <= num - 1; i++)

    {

        if (num % i == 0)

        {

            return false;

        }

        return true;

    }

}

int main()

{

    int a, b;

    cout << "ENter the numbers respectively, from where you want to checking for prime number \n"

         << endl;

    cin >> a >> b;

    for (int i = a; i <= b; i++)

    {

        if (isprime(i))

        {

            cout << i << endl;

        }

    }

}

// \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_

// \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_

    // Time and Space Complexity -

    // int n;

    // cin >> n;

    // // 1 -

    // for (int i = 1; i <= n; i++)

    // {

    //     cout << i << endl;

    // }

    // // O(n)

    // // 2 -

    // for (int i = 1; i <= 2 \* n; i++)

    // {

    //     cout << i << endl;

    // }

    // // O(2n) = O(n)

    // // 3 -

    // for (int i = 1; i <= n; i++)

    // {

    //     cout << i << endl;

    // }

    // for (int j = 1; j <= n; j++)

    // {

    //     cout << j << endl;

    // }

    // // O(n+n) = O(2n)

    // // 4 -

    // for (int i = 1; i <= n; i++)

    // {

    //     for (int j = 1; j <= 5; j++)

    //     {

    //     }

    // }

    // // O(n\*5) = O(n)

    // // 5 -

    // for (int i = 1; i <= n; i++)

    // {

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

    //     {

    //         cout << j << endl;

    //     }

    // }

    // // O(n\*n) = O(n^2)

    // // 6 -

    // for (int i = 1; i <= n; i += 2)

    // {

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

    //     {

    //         cout << j << endl;

    //     }

    // }

    // // O((n/2)\*n = O(n^2))

    // // 7 -

    // for (int i = 1; i <= n; i \*= 2)

    // {

    //     cout << i << endl;

    // }

    // // O(log2n)

    // // 8 -

    // for (int i = 1; i <= n; i \*= 2)

    // {

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

    //     {

    //     }

    // }

    // // O(n\*log2n)

    // // 9 -

    // for (int i = 1; i <= n; i \*= 2)

    // {

    // }

    // for (int j = 1; j <= n; j++)

    // {

    // }

    // // O(log2n + n) = O(n) dominates as bigger than logn

// Space Complexiotyu -

    // int n;

    // cin>>n;

    // // 1 -

    // int arr[n];

    // // O(n)

    // // 2 -

    // int matrix[n][n];

    // // O(n^2)+O(n) = O(n^2)

    // // 3 -

    // int rev = 0;

    // while (n!=0)

    // {

    //     rev = rev+10+n%10;

    //     n/=10;

    // }

    // cout<<rev;

    // // S.C - O(2) - for 2 var n&rev, both are constant = O(1)

    // // T.C - Jitne digits utne complexity. 2 times step - 1) for modular & 2 for division - so O(2\*no. of digits in no.) = O(No. of digits)

// \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_