**01.) PRINT DIVISOR**

class Solution {

  public:

    void print\_divisors(int n) {

        vector<int>divisors;

        for(int i=1 ;i\*i<=n;i++)

        {

            if(n%i==0)

            {

                divisors.push\_back(i);

                if(n/i!=i)

                {

                    divisors.push\_back(n/i);

                }

            }

        }

        sort(divisors.begin(), divisors.end());

        for(int val:divisors)

        {

            cout << val << " ";

        }

    }

};

**02.) CHECK IF NUMBER IS PRIME OR NOT**

**CODE 01**

class Solution{

public:

    int isPrime(int N){

        if (N <= 1) return 0;

        if (N == 2) return 1;

        if (N % 2 == 0) return 0;

        for (int i = 3; i <= sqrt(N); i += 2) {

            if (N % i == 0) return 0;

        }

        return 1;

    }

};

**CODE 02**

class Solution{

public:

    int isPrime(int N){

        if (N <= 1) return 0;

        if (N <= 3) return 1;

        if (N % 2 == 0 || N % 3 == 0) return 0;

        for (int i = 5; i \* i <= N; i += 6) {

            if (N % i == 0 || N % (i + 2) == 0) return 0;

        }

        return 1;

    }

};

**CODE 03**

class Solution{

public:

    int isPrime(int N){

        if (N <= 1) return 0;

        if (N <= 3) return 1;

        for (int i = 2; i <= sqrt(N); i++) {

            if (N % i == 0) return 0;

        }

        return 1;

    }

};

**03.) PRINT PRIME FACTORS**

#include <bits/stdc++.h>

using namespace std;

void primeFactors(int n)

{

    while (n % 2 == 0)

    {

        cout << 2 << " ";

        n = n / 2;

    }

    for (int i = 3; i <= sqrt(n); i = i + 2)

    {

        while (n % i == 0)

        {

            cout << i << " ";

            n = n / i;

        }

    }

    if (n > 2)

        cout << n << " ";

}

int main()

{

    int n = 315;

    primeFactors(n);

    return 0;

}

**04.) SIEVE OF ERATHOTHENES**

#include <bits/stdc++.h>

using namespace std;

void SieveOfEratosthenes(int n) {

    bool prime[n + 1];

    memset(prime, true, sizeof(prime));

    for (int p = 2; p \* p <= n; p++) {

        if (prime[p] == true) {

            for (int i = p \* p; i <= n; i += p)

                prime[i] = false;

        }

    }

    for (int p = 2; p <= n; p++)

        if (prime[p])

            cout << p << " ";

}

int main() {

    int n = 30;

    cout << "NUMBERS ARE :- " << n << endl;

    SieveOfEratosthenes(n);

    return 0;

}

**THANK YOU !**