# PROIECT NR. 2 : NUMERE PRIME

A elaborat: Corina CERNOLEV, gr. IS11Z

A verificat: Vitalie ŢÎCĂU, lector universitar

## 1. Formularea problemei

1. De verificat dacă un număr este prim;
2. De indicat toate numerele prime din intervalul *[a; b]*;
3. De indicat primele *n* numere prime;
4. De indicat toate numerele prime inverse din intervalul *[a; b]*;
5. De determinat cel mai mare număr prim.

## 2. Studiu preliminar

Un număr prim este un [număr natural](https://ro.wikipedia.org/wiki/Num%C4%83r_natural), mai mare decât 1, care are exact doi [divizori](https://ro.wikipedia.org/wiki/Divizor) pozitivi: numărul 1 și numărul în sine. Primul număr prim este numărul 2.

## 3. Programul

#include <bits/stdc++.h>

using namespace std;

bool isPrime(long n)

{if (n == 2 || n == 3 || n == 5){

        return true;}

    else if (n % 2 == 0 || n % 3 == 0 || n % 5 == 0){

        return false;}else{

        int i = 7;

        while (i <= sqrt(n)){

            if (n % i == 0){

                return false; }

            i += 2; }}

    return true;}

// bool Prim(int n)

// {if (n < 2) return false;

//     for (int k = 2; k <= n / 2; k++)

//         if (n % k == 0) {return false;}

//     return true; }

long Invers(long n){ long r = 0;

    while (n > 0) {

        r = r \* 10 + n % 10;

        n = n / 10; }return r;}

int main()

{cout << "Proiect nr. 2 elaborat de Cernolev Corina\n";

    cout << "Tema: Numere prime\n";

    int numar, n;

    long lsus, ljos;

    // De verificat daca numarul este prim

    do{

        cout << "1.\tDati un numar natural: ";

        cin >> numar; } while (numar < 2);

    if (isPrime(numar)) cout << "\tNumarul " << numar << " este prim\n";

    else cout << "\tNumarul " << numar << " nu este prim\n";

    cout << "2.\tDati intervalul\n";

    do{

        cout << "\tLimita inferioara: ";

        cin >> ljos;} while (ljos < 2);

    cout << "\tLimita superioara: ";

    cin >> lsus;

     cout << "\tNumerele prime din intervalul [" << ljos << ";" << lsus << "] sunt: \n";

    int nr=0;

    for (int i = ljos; i <= lsus; i++){

        if (isPrime(i)){

            cout << setw(9) << i;

            nr++;}}

    cout << "\nIn total " << nr << " numere prime\n";

    cout << "\tNumerele prime inverse din intervalul [" << ljos << ";" << lsus << "] sunt: \n";

    nr = 0;

    for (long i = ljos; i <= lsus; i++)

    {if (isPrime(i) && isPrime(Invers(i)))

        {cout << setw(9) << i;

            nr++; }}

    cout << "\nIn total " << nr << " numere prime inverse\n";

    // Primele n numere prime

    do{cout << "\n4.\tIntroduceti nr de nr prime: ";

        cin >> n; } while (n < 1);

    int k = 2;

    cout << "\tPrimele " << n << " numere prime sunt: \n";

    while (n > 0){

        if (isPrime(k))

        {cout << setw(9) << k;

            n--;}

        k++;}

    cout << "\nAcestea sunt primele numere prime\n";

    // Cel mai mare numar prim

    long i = 2147483647;

    while (i > 2)

    {if (isPrime(i)) {

            cout << "\n5.\t" << i << " este cel mai mare numar prim";

            break; }

        i--;}

    return 0;}

## 4. Concluzii

Cel mai mare numar prim este 2147483647.

Numerele prime sunt mai numerice decât numerele prime inverse.

Există două metode de verificare dacă numărul este prim: prin parcurgerea până la radical din n și până la n/2. Este mai eficientă metoda parcurgerii până la radical din n.