**A elaborat: Davidov Ecaterina, grupa IS11Z**

**A verificat: Vitalie ȚICĂU, lector univ.**

# PROIECT DE PROGRAME NR. 2. ORGANIZAREA CICLURILOR ÎN PROGRAM. FUNCŢII

Tema: Numerele Armstrong

## I. Formularea problemei

De efectuat următoarele operaţii conform variantei:

1. De verificat dacă numărul *nr* satisface proprietatea de bază.
2. De indicat toate numerele din intervalul *[a; b]*, care satisfac proprietatea de bază.
3. De indicat primele *n* numere, care satisfac proprietatea de bază.
4. De indicat toate numerele din intervalul *[a; b]*, care satisfac proprietatea de bază (varianta optimizată) sau care satisfac a doua proprietate.
5. De determinat cel mai mare număr, care satisface condiţia indicată.

## II. Numerele Armstrong

*Numărul* natural *n* din *k* cifre este un *număr Armstrong* (uneori, *număr narcisist*) dacă este egal cu suma cifrelor sale ridicate la puterea *k*. Primele 21 numere Armstrong: 1, 2, 3, 4, 5, 6, 7, 8, 9, 153, 370, 371, 407, 1634, 8208, 9474, 54748, 92727, 93084, 548834, 1741725. Există doar 4 numere Armstrong cu trei cifre: 153=13+53+33; 370=33+73+03; 371=33+73+13 şi 407=43+03+73. De fapt întreaga mulţime a numerelor Armstrong cuprinde doar 88 de termeni, întrucât s-a demonstrat că nu pot exista astfel de numere cu mai mult de 60 de cifre. Dintre aceste 88 de numere, 7 sunt prime. Cele 7 prime Armstrong: 2, 3, 5, 7, 28116440335967, 449177399146038697307, 35452590104031691935943

## III. Programul

## #include <iostream>

## using namespace std;

## /\*

## 1.Calculam numarul de cifre

## 2.Verificam proprietatea

## \*/

## int NrCifre(long a)

## {

## int kol = 0;

## while (a > 0)

## {

## a /= 10;

## kol++;

## }

## return kol;

## }

## long power(long a, long b)

## {

## long prod = 1;

## for (long i = 1; i <= b; i++)

## {

## prod \*= a;

## }

## return prod;

## }

## bool Armstrong(long nr)

## {

## long sum = 0, cf, col = NrCifre(nr);

## long number = nr;

## while (number > 0)

## {

## cf = number % 10;

## sum += power(cf, col);

## number /= 10;

## }

## return (sum == nr);

## }

## int main()

## {

## cout << "Proiect nr.2 elaborat de Davidov Ecaterina" << endl;

## cout << "Tema: Numere Armstrong" << endl;

## long nr;

## cout << "Introduceti un numar: ";

## cin >> nr;

## Armstrong(nr) ? cout << nr << " este numar Armstrong" : cout << nr << " nu este numar Armstrong";

## return 0;

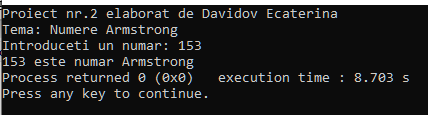
## }

## \*/IV. Rezultatele obținute

## Proiect Nr.2 elaborat de Davidov Ecaterina

## Tema: Numerele Armstrong

## indica valoarea unui număr natural care într-un sistem de numere dat este egală cu suma cifrelor sale ridicate la o putere egală cu numărul cifrelor sale.



## V. Concluzii

Folosind acest algoritm putem calcula valoarea unui număr natural care a fost ridicat la o putere în timpul minim. Dacă numărul introdus în program nu se potrivește cu parametrii, programul dă răspunsul „Nu este un număr Armstrong”