**A elaborat: Ulmada Dmitri, grupa IS11Z**

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

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

# Tema: Numerele Lucas

## 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. Studiul preliminar

*Numerele Lucas* sunt definite prin relaţia de recurenţă *Ln = Ln-1+Ln-2*, unde *L0=1* şi *L1=3*. O altă formulă pentru calculul celui de-al n-lea număr Lucas este: *Ln = ((1+√5)/2)n + ((1–√5)/2)n*. Primele 20 numere Lucas: 1, 3, 4, 7, 11, 18, 29, 47, 76, 123, 199, 322, 521, 843, 1364, 2207, 3571, 5778, 9349, 15127. Un număr Lucas ce este totodată şi prim se numeşte *prim Lucas*. Primele 16 prime Lucas: 2, 3, 7, 11, 29, 47, 199, 521, 2207, 3571, 9349, 3010349, 54018521, 370248451, 6643838879, 119218851371.

## III. Programul

#include <iostream>

#include <stdlib.h>

#include <math.h>

#include <iomanip>

#define NMAX 30

using namespace std;

int lucas(int n);

int main()

{

int numar;

int i;

int jos, sus;

int contor, cit;

cout << "Proiect Nr. 2, elaborat de Ulmada Dmitri" << endl;

cout << "Tema: Numerele Lucas\n";

cout << "Introduceti numarul intreg, pentru verificare: ";

cin >> numar;

cout << "Introduceti limitele de jos si de sus: ";

cin >> jos >> sus;

cout << "Cite numere lucas: ";

cin >> cit;

int temp = 0;

int cont = 0;

bool check = false;

for(cont = 0; cont <= numar; cont++)

{

temp = lucas(cont);

if(temp == numar)

{

cout << "\n1. " <<numar <<" este numarul Lucas";

check = true;

break;

}

else

cont++;

}

if (!check)

cout << "\n1. " << numar <<"\ nu este lucas";

cout << "\n2. Numerele lucas de la " << jos << " la " << sus << endl;

i=1;

while (lucas (i) >jos && lucas (i) <sus)

{

cout << setw(8) << lucas (i);

i++;

}

cout << "\n3. Primele "<< cit << " numere lucas " << endl;

for( contor = 0; contor < cit; contor++)

cout << setw(8) << lucas(contor) ;

if(contor)

cout << "\nIn total: " << contor << " numere\n";

else

cout << "Nu sunt numere lucas pe interval";

cout << "\n4. Numere lucas sunt:\n";

jos = 1;

sus = 10;

for (numar = jos; numar <= sus; numar++)

{

cout << setw(8) << lucas(numar);

}

cout << "\n5. Cel mai mare numar lucas este: ";

cout << lucas(NMAX);

return 0;

}

int lucas(int n)

{

if (n == 0)

return 1;

if (n == 1)

return 3;

return lucas(n - 1) +

lucas(n - 2);

}

## IV. Rezulatatele obţinute

Proiect Nr. 2, elaborat de Ulmada Dmitri

Tema: Numerele Lucas

Introduceti numarul intreg, pentru verificare: 41

Introduceti limitele de jos si de sus: 1 100

Cite numere lucas: 15

1. 41 nu este lucas

2. Numerele lucas de la 1 la 100

3 4 7 11 18 29 47 76

3. Primele 15 numere lucas

1 3 4 7 11 18 29 47 76 123 199 322 521 843 1364

In total: 15 numere

4. Numere lucas sunt:

3 4 7 11 18 29 47 76 123 199

5. Cel mai mare numar lucas este: 3010349

## V. Concluzii

1. Programul aplică corect formula pentru cifre lucas;
2. Programul este capabil să calculeze dacă cifra de intrare este un număr lucas și arată N numărul de numere lucas în intervalul dat.