**A elaborat:Nicolai Russu, grupa IS11Z**

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

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

# Tema: Numere Pell

## **II. Formularea problemei**

1. De verificat dacă număruleste Pell.
2. De indicat toate numerele Pelle din intervalul *[a; b]*, care satisfac proprietatea de bază.
3. De indicat primele *n* numere Pell.
4. De indicat toate numerele Pelle din intervalul *[a; b]*, varianta optimizată
5. De determinat cel mai mare număr Pell

## **II. Studiul preliminar**

*Numerele Pell* sunt definite prin recurenţă: *Pn = 2Pn-1+Pn-2*, unde *P0=1* şi *P1=1*

## **III. Programul**

#include <iostream>

#include <stdlib.h>

#include <math.h>

#include <iomanip>

#define NMAX 1500000

using namespace std;

int Pell(int n);

bool Prim(int n);

int main()

{

int numar, jos, sus, contor, cit, n , s;

cout << "Proiect Nr. 2, elaborat de Russu Nicolai" << endl;

cout << "Tema: Numere Pell\n";

//1. De verificat dacă numărul nr satisface proprietatea de bază.

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

cin >> numar ;

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

cin >> jos >> sus;

cout << "Cite numere Pell: ";

cin >> cit;

if (Pell(numar))

cout << "\n1. Numarul " << numar << " este Pell" << endl;

else

cout << "\n1. Numarul " << numar << " nu este Pell" <<endl;

//2. De indicat toate numerele din intervalul [a; b], care satisfac proprietatea de bază.

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

for(numar = jos, contor = 0; numar <= sus; numar++)

if (Pell(numar))

{

cout << setw(8) << numar;

contor++;

}

if(contor)

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

else

cout << "Nu sunt numere Pell pe interval\n";

//3. De indicat primele n numere, care satisfac proprietatea de bază.

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

for(numar = 1, contor = 0; contor < cit; numar++)

if (Pell(numar))

{

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

contor++;

}

if(contor)

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

else

cout << "\nNu sunt numere Pell pe interval\n";

//4. De indicat toate numerele din intervalul [a; b], care satisfac proprietatea de bază (varianta optimizată) sau care satisfac a doua proprietate.

cout << "\n4. Numere Pelle si Prime:\n";

for(numar = jos, contor = 0; numar <= sus; numar++)

if (Pell(numar)) {

if (Prim(numar)){

cout << numar <<endl; contor++;

}

}

if(contor == 0)

cout << "Nus numere Pelle si Prime"<<endl

//5. De determinat cel mai mare număr, care satisface condiţia indicată.

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

for (numar = NMAX; numar > 0; numar--)

{

if (Pell(numar))

{

cout << numar << endl;

break;

}

}

return 0;

}

int Pell(int n)

{

int a = 0;

int b = 1;

int s;

s = a + b \* 2;

while(s < n)

{

a = b;

b = s;

s = a + b \* 2;

}

return s == n;

}

bool Prim(int n)

{

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

if (n % i == 0) return false;

return true;

}

## **IV. Rezulatatele obţinute**

Proiect nr.2 elaborat de Russu Nicolai

Tema: Numere Pell

Introduceti numarul pentru derificare: 10

Introduceti limitele de jos si de sus: 1 – 12005

1. Numarul 10 nu este Pell
2. Numere Pell de la 1 la 12005

2 5 12 29 70 169 408 985 2378 5741

1. Primele 10 numere Pell

2 5 12 29 70 169 408 985 2378 5741

In total 10 numere

1. Numere Pelle si Prime: 2 5 29 5741
2. Cel mai mare numar Pell este : 1136689

## **V. Concluzii**

Нумере пел можно вычислять по обычной формуле и оптимизированой оптимизированная формула считается быстрей