Національний технічний університет України

«Київський політехнічний інститут імені Ігоря Сікорського»

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

Алгоритми та методи обчислення

Лабораторна робота №1

**«Поняття алгоритму. Задавання алгоритмів у вигляді блок-схем»**

Виконала:

студентка групи ІВ-71

Молчанова В.С.

Перевірив:

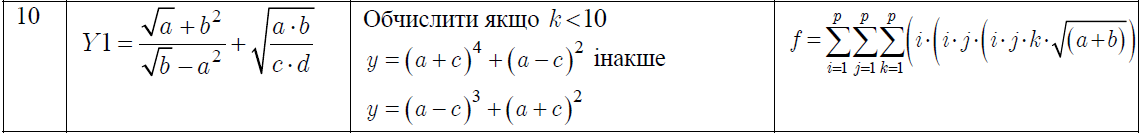
ст.вик. Порєв В. М.

Київ

2019 р.

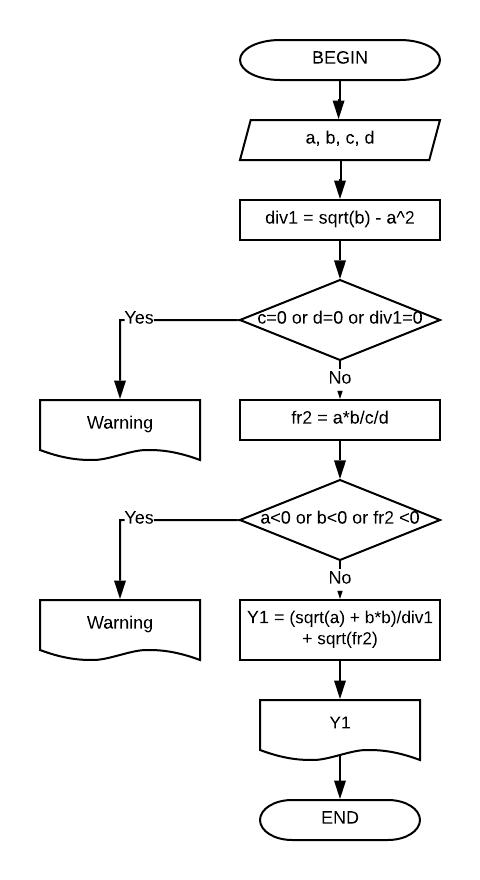
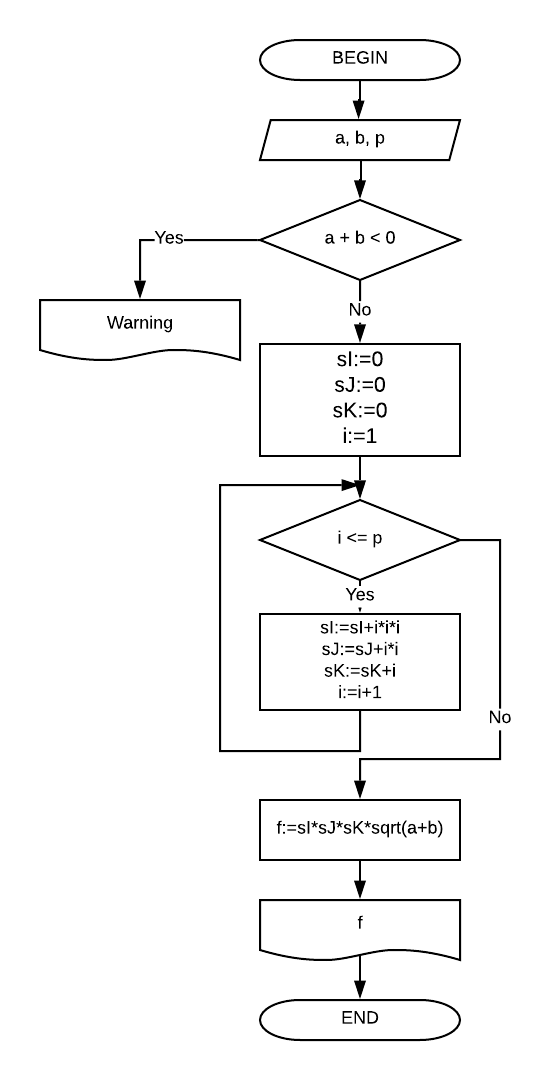
**Мета:** Навчитися створювати блок-схеми лінійного алгоритму; розгалуженого алгоритму та циклічного алгоритму за допомогою редактора блок-схем *afce*або іншого довільного редактора.

**Завдання:** Відповідно до варіанту завдання розробити блок-схеми обчислення виразів для лінійного алгоритму, алгоритму, що розгалужується та циклічного алгоритму. У відповідності до блок-схеми створити програму обчислення виразу алгоритмічною мовою, узгодженою з викладачем.

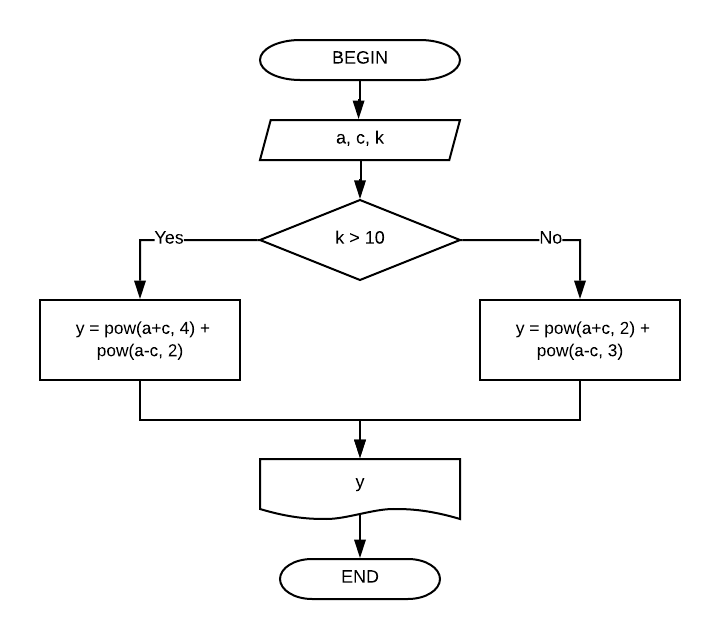


**Блок-схеми алгоритмів**

Лінійний алгоритм: Циклічний алгоритм:



Алгоритм з розгалудженням:



**Спрощення виразу циклічного алгоритму:**



**Роздруківка тексту програми:**

#include "appcore.h"

#include <math.h>

AppCore::**AppCore**(QObject \*parent) : QObject(parent)

{

}

void AppCore::**receiveLinear**(double a, double b, double c, double d){

calculateLinear(a, b, c, d);

}

void AppCore::**receiveBranching**(double a, double c, double k){

calculateBranching(a, c, k);

}

void AppCore::**receiveCyclic**(double a, double b, int p){

calculateCyclic(a, b, p);

}

void AppCore::**calculateLinear**(double a, double b, double c, double d){

double divider1 = sqrt(b) - a\*a;

if(c == 0 || d == 0 || divider1 == 0){

emit sendLinearWarning("Нельзя делить на ноль!");

return;

}

double fraction2 = a \* b / c / d;

if(a < 0 || b < 0 || fraction2 < 0){

emit sendLinearWarning("Нельзя извлекать корень из отрицательного числа!");

return;

}

emit sendLinear((sqrt(a) + b\*b)/divider1 + sqrt(fraction2));

}

void AppCore::**calculateBranching**(double a, double c, double k){

if(k < 10){

emit sendBranching(pow(a+c, 4) + pow(a-c, 2));

return;

}

emit sendBranching(pow(a+c, 2) + pow(a-c, 3));

}

void AppCore::**calculateCyclic**(double a, double b, int p){

if(a + b < 0){

emit sendCyclicWarning("Нельзя извлекать корень из отрицательного числа!");

return;

}

double sumI = 0, sumJ = 0, sumK = 0;

for(int i = 1; i <= p; i++){

sumI += i\*i\*i;

sumJ += i\*i;

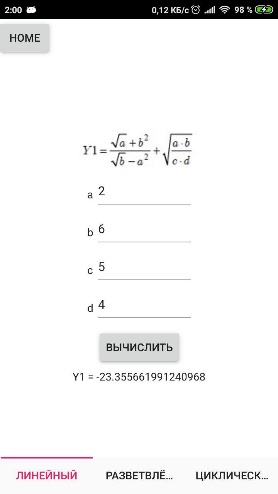
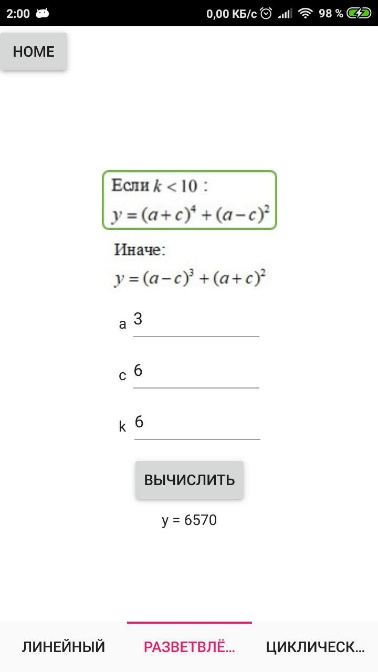
sumK += i;

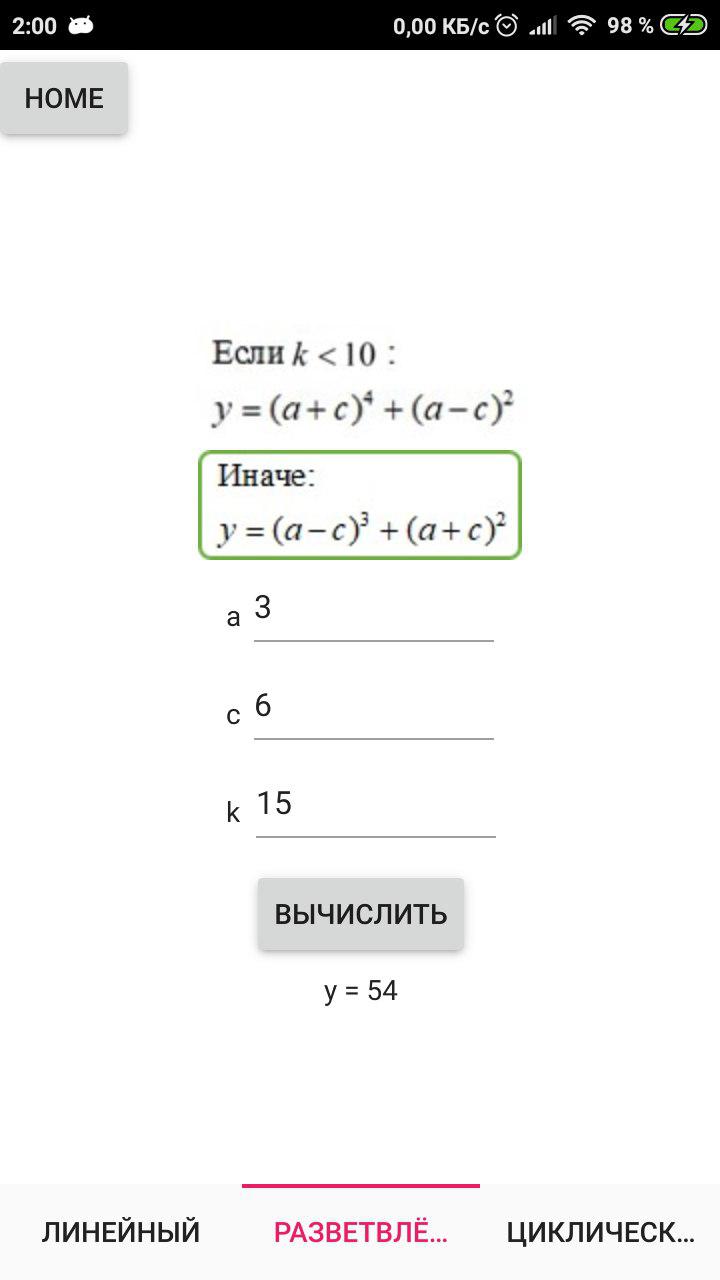
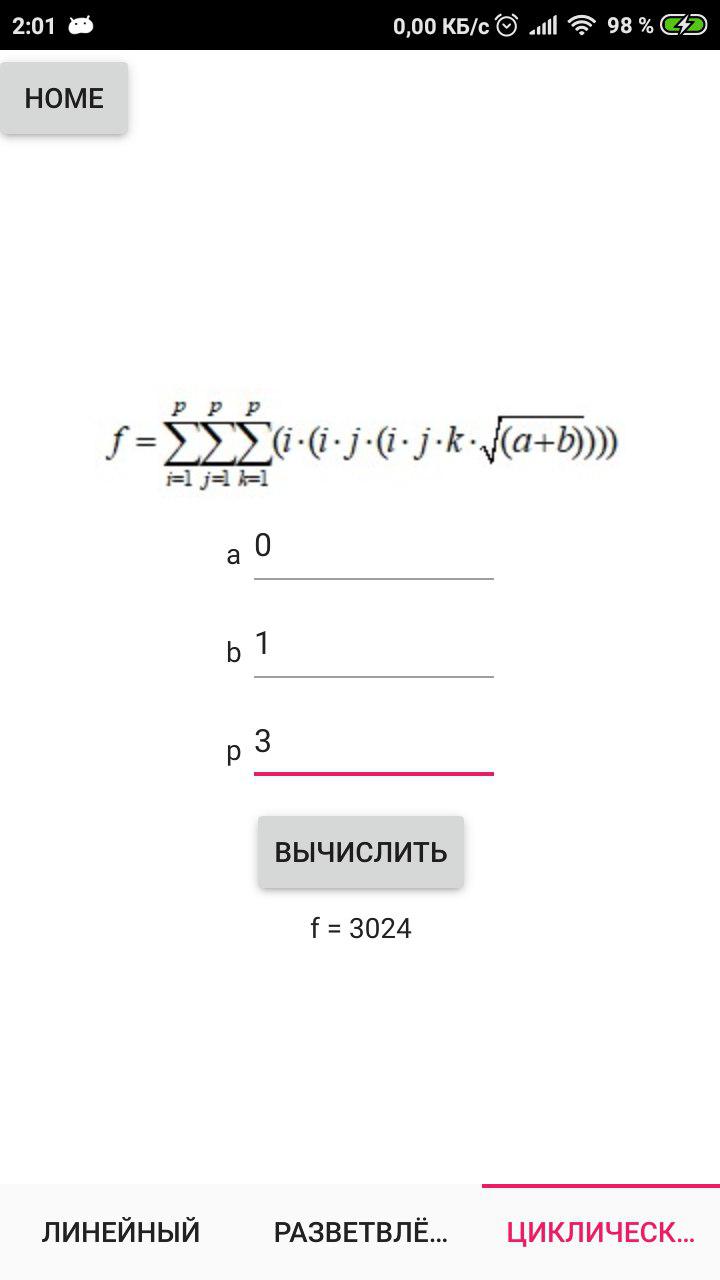
}

emit sendCyclic(sqrt(a+b)\*sumI\*sumJ\*sumK);

}

**Роздруківки результатів виконання програми**

****

****

**Висновки:** У ході виконання лабораторної роботи я закріпила знання з базових понять алгоритмів, вивчила основні правила складання блок-схем алгоритмів і покращила навички роботи з MS Visio. Отримані результати виконання програми є вірними.