 МІНІСТЕРСТВО ОСВІТИ I НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ   
ІМЕНІ ІГОРЯ СІКОРСЬКОГО»

ФАКУЛЬТЕТ БІОМЕДИЧНОЇ ІНЖЕНЕРІЇ

КАФЕДРА БІОМЕДИЧНОЇ КІБЕРНЕТИКИ

**Комп’ютерний практикум №4**

з дисципліни «Алгоритмізація та програмування»

на тему: «Алгоритми розгалуженої структури(умовний оператор)»

Варіант №2

**Виконав:**

студент гр. БС-71

Батуркіна А.М.

**Перевірив:**

ас. каф. БМК

Рисін С. В.

Зараховано від \_\_\_.\_\_\_.\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(підпис викладача)

Київ-2018

**🞏 Комп’ютерний практикум без зауважень**

**🞏 Комп’ютерний практикум має зауваження:**

**🞏 несвоєчасний захист**

**🞏 присутні зауваження до блок-схеми:**

**🞏 блок-схема не відповідає коду**

**🞏 в блок-схемі присутній код**

**🞏 виконані не за стандартом:**

**🞏 блок умови 🞏 визначений процес (функція)**

**🞏 оператор вибору 🞏 перехід**

**🞏 цикл 🞏 розміри блоків**

**🞏 інші зауваження:**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**🞏 присутні зауваження до коду:**

**🞏 задача завдання вирішена хибно**

**🞏 код програми не компілюється**

**🞏 використано глобальні змінні**

**🞏 типи даних визначені хибно**

**🞏 недостатня декомпозиція на функції користувача**

**🞏 функція main містить лише виклик іншої функції**

**🞏 статичні змінні при роботі з масивами**

**🞏 оформлення коду**

**🞏 присутні зайві символи «{» та «}»**

**🞏 інші зауваження:**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**🞏 результати виконання програми на рисунках не відповідають коду**

**🞏 невірні відповіді на запитання:**

**🞏 №1 🞏 №2 🞏 №3 🞏 №4 🞏 №5**

**🞏 №6 🞏 №7 🞏 №8 🞏 №9 🞏 №10**

**🞏 незнання теоретичного матеріалу**

**🞏 маються інші зауваження:**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

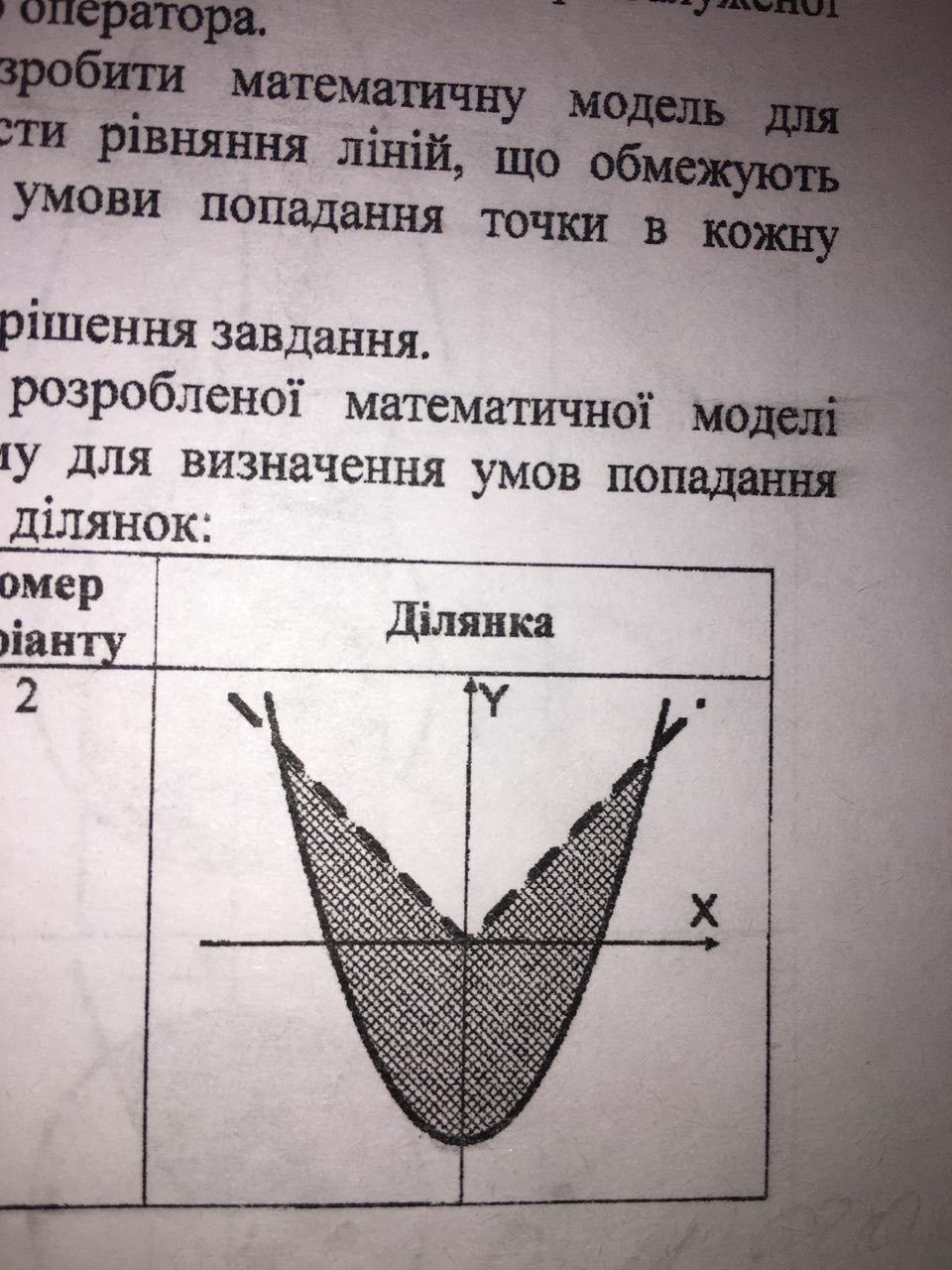
**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Завдання:**

1. Вивчити теоретичні основи написання алгоритмів розгалуженої структури з використанням умовного оператора.
2. Відповідно до свого варіанту розробити математичну модель для визначення ділянки площини: навести рівняння ліній, що обмежують заштриховані ділянки, та описати умови попадання точки в кожну ділянку.



1. Побудувати блок-схему алгоритму вирішення задання.
2. Відповідно до свого варіанту та розробленої математичної моделі розбиття площини написати програму для визначення умов попадання заданої користувачем точки в кожну з ділянок.
3. Скласти та захистити звіт по роботі.

**Рівняння ліній, що обмежують заштриховані ділянки:**

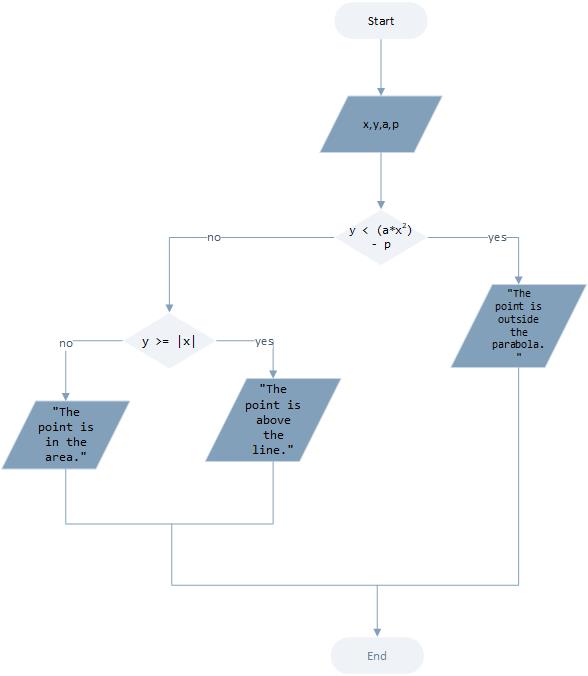
*1) y=a\*x2-p*

*2) y=|x|*

*(точки, які потрапляють на лінію рівняння №2, вважаються поза ділянкою)*

**Умови попадання точки в кожну ділянку: **

**Блок-схема:**

****

**Лістинг:**

#include <iostream>

#include <conio.h>

using namespace std;

int main()

{

float x, y, p, a;

cout << "Input x: ";

cin >> x;

cout << "Input y: ";

cin >> y;

cout << "Input a: ";

cin >> a;

cout << "Input p: ";

cin >> p;

if (y < (a\*x\*x) - p)

cout << "The point is outside the parabola." << '\n';

else

{

if (y >= fabs(x))

cout << "The point is above the line.";

else

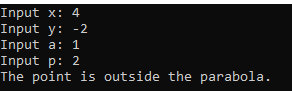
cout << "The point is in the area.";

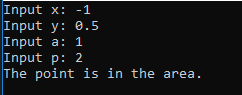
}

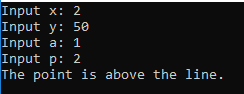
\_getch();

}

**Результат:**



****

****

**Контрольні питання:**

1. Що таке алгоритм розгалуженої структури?

* Алгоритм розгалуженої структури – це вид керуючої структури, в якому деякі дії виконуються(або не виконуються) в залежності від деяких умов.

1. Визначити синтаксис умовного оператора.

* Формат скороченої форми:

if(expression)

statement;

де expression – умова прийняття рішення, statement - простий або складений оператор.

Формат повної форми:

1)if(expression)

statement1;

else

statement2;

2)a=(B)? S1:S2;

B-умова, яка перевіряється на істинність

S1-виконується, коли умова істинна

S2-виконується, коли умова хибна

1. Яка різниця між умовними операторами з однією та двома гілками?

* Умовний оператор повної форми(з else) залежно від результату перевірки дозволяє виконати ту чи іншу дію(тобто має дві гілки). Умовний оператор скороченої форми(без else) дозволяє виконати дію лише при виконанні умови(тобто має одну гілку).

1. Чи можливе існування тільки гілки else(без операторів if–умови) в умовному операторі?

* Ні, існування тільки гілки else не можливе, так як гілка else буде виконуватись, лише коли умова при перевірці не справдилась(хибна).

1. Коли в умовному операторі слід використовувати фігурні дужки «{» та «}», а коли ні?

* Якщо, після виконання умови, потрібно виконати декілька операторів, то ці оператори беруться у фігурні дужки:

1. Що таке логічний вираз?

* Логічний вираз – це конструкція мови програмування, результат обчислення якої може бути істинним(тобто значення логічного виразу дорівнює true(не «0»)) або хибним(тобто значення логічного виразу дорівнює false(«0»)).

1. Перелічіть логічні операції, що використовуються в логічних виразах при написанні програм мовою С++.

* 1)заперечення(«!»); 2)логічне додавання або логічне «АБО»( «||»); 3)логічне множення або логічне «І»( «&&»).

1. Яка операція («&&» чи «||») має більший пріоритет? Навести відповідні приклади для пояснення відповіді.

* Оператор логічного І («&&)» має більший пріоритет, ніж оператор логічного АБО («||»).

Приклад: expr1 || expr2 && expr3 – у даному випадку, так як пріоритет І «&&» більше, то: expr1 || (expr2 && expr3)

(те, що в дужках, обробляється першим)

1. Як будується складна умова, що об’єднує декілька умов?

- Складна умова може бути побудована завдяки логічним операціям І«&&»(тобто логічна кон’юнкція) та АБО«||»(тобто логічна диз’юнкція):

1)if (exp1&&exp2)

2)if(exp1||exp2)

1. Чи виконається в наступному коді одна з гілок умовного оператора?

int x = 0, y = -1;

if (x = 0)

y = x;

else

y = 1;

cout << y;

Відповідь пояснити та визначити, що буде виведено на екран.

В даному коді не виконається гілка if , тому, що в умові написано присвоїти «х» значення 0, результатом чого є зміна значення змінної. Через це в if потрапляє 0,що є еквівалентно логічному значенню false. Саме тому виконується лише гілка else.