Міністерство освіти і науки України

Національний університет «Львівська політехніка»

Кафедра систем штучного інтелекту



**ЗВІТ**

про виконання

**Розрахункової роботи**

***з дисципліни:*** «Мови та парадигми програмування»

**Затверджено**

на засіданні кафедри

систем штучного інтелекту

Протокол № \_\_ від \_\_\_\_\_\_\_\_\_\_

***Виконала:***

студентка групи ШІ-13

Фесенко Дар’я Максимівна

# **Тема роботи:**

Графічна розрахункова робота.

# **Мета роботи:**

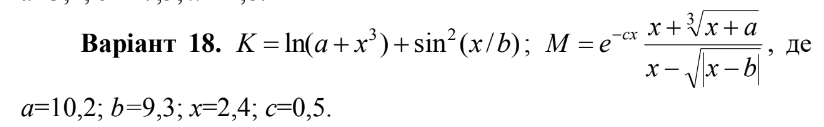
Навчитися розв’язувати основні математичні задачі за допомогою мови програмування C++/C.

# **Виконання роботи:**

## **1. Опрацювання завдання та вимог до програм та середовища:**

Завдання №1 VNS Practice Work 1.

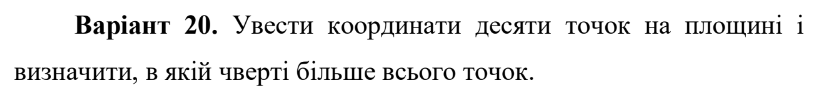
**Деталі завдання:** розробити лінійний алгоритм для розв’язання задачі.



**Важливі деталі для врахування в імплементації програми**: правильний запис формул із урахуванням порядку виконання операцій.

Завдання №2 VNS Practice Work 2.

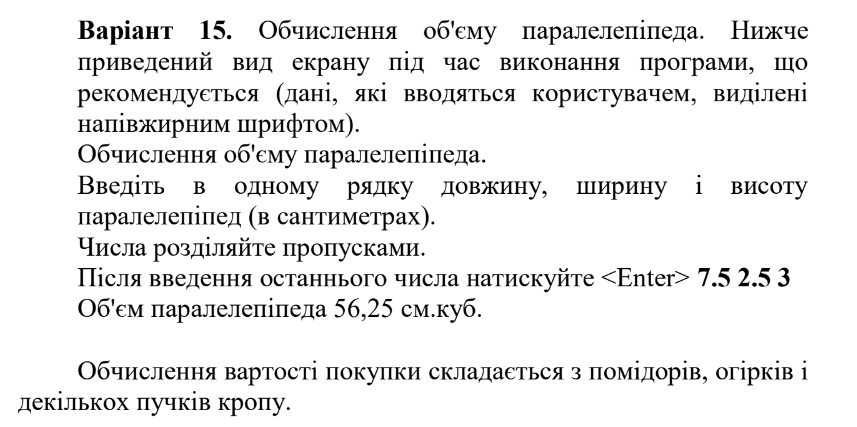
**Деталі завдання:** розробити алгоритм, що розгалужується для розв’язання задачі номер якої відповідає порядковому номеру студента в журналі викладача.



**Важливі деталі для врахування в імплементації програми**: врахувати умови визначення чверті, враховуючи від'ємні та додатні значення координат.

Завдання №3 VNS Practice Work 3.

**Деталі завдання:** написати програму згідно свого варіанту.



**Важливі деталі для врахування в імплементації програми**: обробити дані та виконати відповідне математичне обчислення об'єму з урахуванням формули.

Завдання №4 VNS Practice Work 4.

**Деталі завдання:** написати програму згідно свого варіанту.



**Важливі деталі для врахування в імплементації програми**: забезпечити введення користувачем цілочисельного значення.

## **2. Дизайн та планована оцінка часу виконання завдань:**

Завдання №1 VNS Practice Work 1.

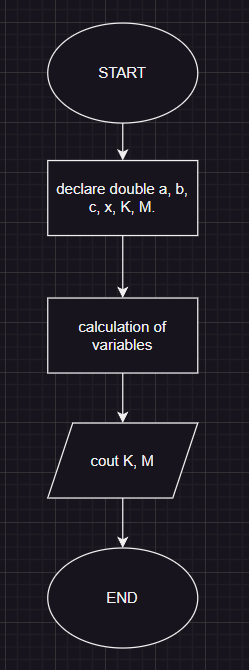


Рисунок Блок-схема до програми 1

Завдання №2 VNS Practice Work 2.

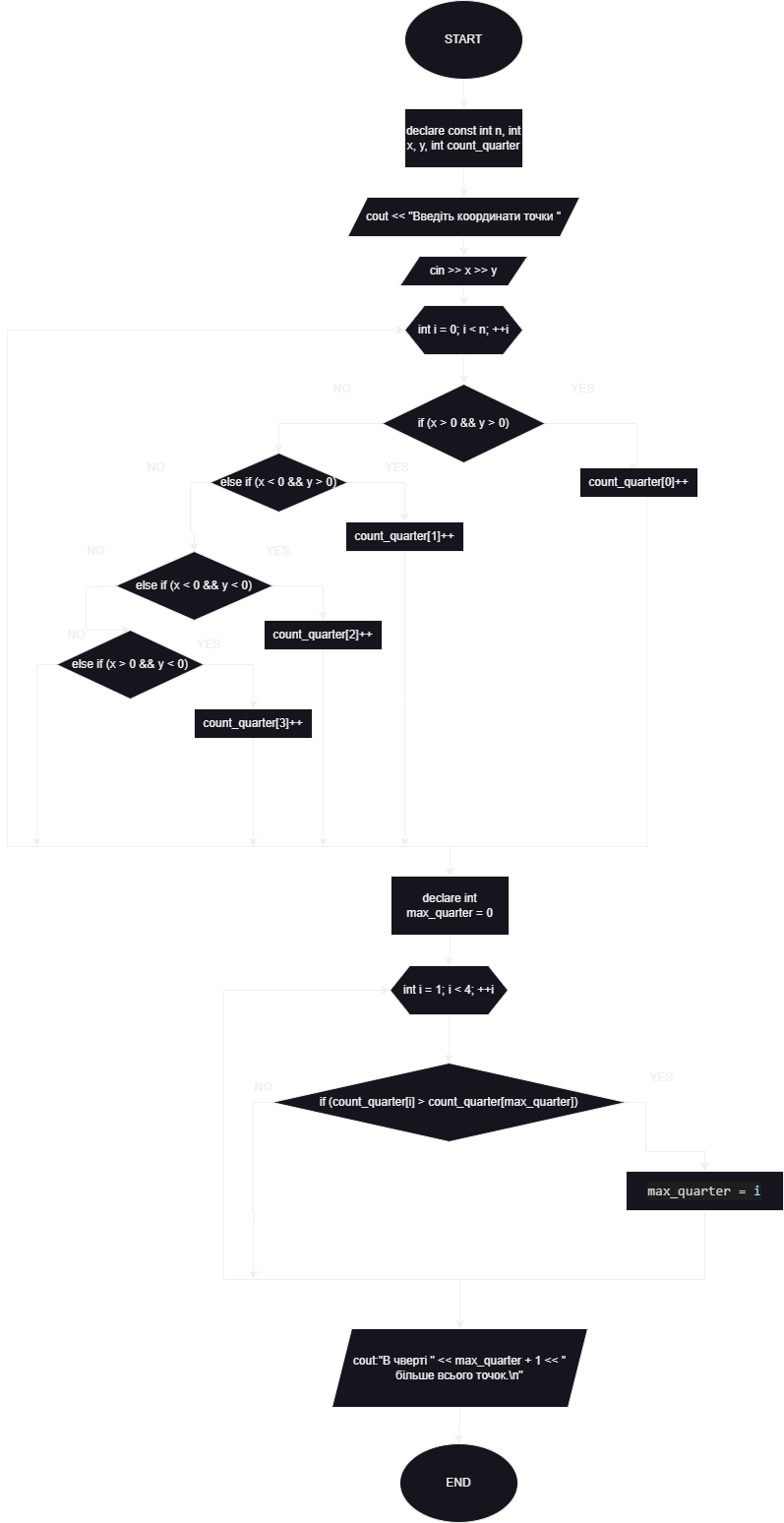


Рисунок Блок-схема до програми 2

Завдання №3 VNS Practice Work 3.

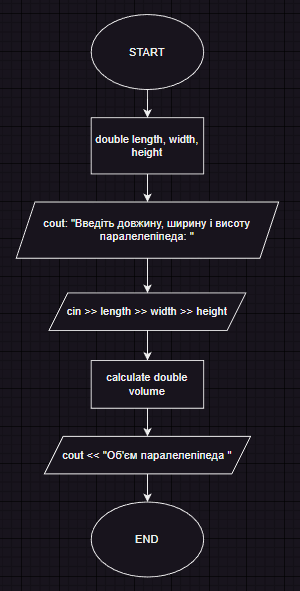


Рисунок Блок-схема до програми 3

Завдання №4 VNS Practice Work 4.

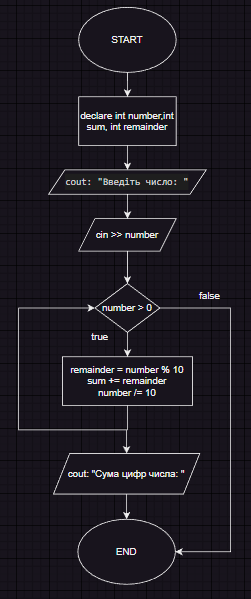


Рисунок Блок-схема до програми 4

**3. Код програм з посиланням на зовнішні ресурси:**

**Pull Request:**

Завдання №1 VNS Practice Work 1.

#include <iostream> // підключення бібліотеки для введення/виведення

#include <cmath> // підключення бібліотеки для математичних обчислень

using namespace std; // використання простору std для уникнення повторень в коді

int main() { // початок головної ф-ції

    double a = 10.2;  // оголошення та ініціалізація типу змінних double, використано дійсні змінні

    double b = 9.3;

    double x = 2.4;

    double c = 0.5;

    double K = log(a + pow(x, 3)) + pow(sin(x / b), 2); //обчислення значення К, використано математичні операції у виразах log, pow, sin і тд.

    double M = exp(1) \* pow(c \* x, 3) \* ((x + cbrt(x + a)) / (x - sqrt(fabs(x - b)))); // обчислення значення M, використано математичні операції у виразах log, pow, sin і тд.

    cout << "K = " << K << endl; // виведення значення К на екран, використано використано оператор вивеення cout

    cout << "M = " << M << endl; // виведення значення M на екран, використано використано оператор вивеення cout

    return 0; // завершення головної ф-ції, повернення значення 0.

}

Завдання №2 VNS Practice Work 2.

#include <iostream> // підключення бібліотеки для введення/виведення

using namespace std; // використання простору std для уникнення повторень в коді

int main() { // початок головної ф-ції

    const int n = 10; // оголошення та ініціалізація константи, використано цілочисельну константу

    int x, y; // оголошення та ініціалізація типу змінних int, використано цілочисельні змінні

    int count\_quarter[4] = {0}; //створено масив для підрахунку кількості точок у кожній чверті, використано цілочисельні змінні

    for (int i = 0; i < n; ++i) { //використано цикл for для введення координат 10 точок

        cout << "Введіть координати точки " << i + 1 << " (x y): "; // повідомлення для ввеення координат точок, використано оператор вивеення cout

        cin >> x >> y; //введення координат, використано оператор вивеення cin

        if (x > 0 && y > 0) { // визначення чверті та збільшення лічильника відповідної чверті, використано умовний оператор if

            count\_quarter[0]++; // перша чверть

        } else if (x < 0 && y > 0) {

            count\_quarter[1]++; //друга чверть

        } else if (x < 0 && y < 0) {

            count\_quarter[2]++; //третя чверть

        } else if (x > 0 && y < 0) {

            count\_quarter[3]++; //четверта чверть

        }

    }

    int max\_quarter = 0;  // знайдення чверті з найбільшою кількістю точок, використано цілочисельні змінні

    for (int i = 1; i < 4; ++i) { // використано цикл for для знаходження найбільшої чверті

        if (count\_quarter[i] > count\_quarter[max\_quarter]) { //використано умовний оператор if для порівняння к-ті точок в чвертях

            max\_quarter = i; // оновлення індексу чверті з найбільшою кількістю точок

        }

    }

    cout << "В чверті " << max\_quarter + 1 << " більше всього точок.\n"; // виведення результату, використано оператор виведення

    return 0; // завершення головної ф-ції, повернення значення 0.

}

Завдання №3 VNS Practice Work 3.

#include <iostream> // підключення бібліотеки для введення/виведення

using namespace std; // використання простору std для уникнення повторень в коді

int main() {  // початок головної ф-ції

    double length, width, height;  // оголошення та ініціалізація типу змінних double, використано дійсні змінні

    // Запитуємо користувача ввести дані

    cout << "Введіть довжину, ширину і висоту паралелепіпеда: "; // виведення запиту для користувача на екран, використано оператор вивеення cout

    cin >> length >> width >> height;  //введення довжини, ширини та висоти паралелепіпеда, використано оператор вивеення cin

    double volume = length \* width \* height; //обчислення об'єму паралелепідела, використано дійсні змінні

    cout << "Об'єм паралелепіпеда " << volume << " см.куб." << endl;  //виведення результату на екран, використано оператор вивеення cout

    return 0; // завершення головної ф-ції, повернення значення 0.

}

Завдання №4 VNS Practice Work 4.

#include <iostream> // підключення бібліотеки для введення/виведення

using namespace std; // використання простору std для уникнення повторень в коді

int main() { // початок головної ф-ції

    cout << "Введіть число: "; // повідомлення для ввеення чисел, використано оператор вивеення cout

    int number; // оголошення та ініціалізація типу змінних int, використано цілочисельні змінні

    cin >> number; //введення чисел, використано оператор вивеення cin

    int sum = 0; // оголошення та ініціалізація типу змінних int, використано цілочисельні змінні

    int remainder;

    while (number > 0) {  //використано цикл while для обчислення суми цифр числа

        remainder = number % 10; // отримання останньої цифри числа

        sum += remainder; //додавання цифри до суми

        number /= 10; //видалення останньої цифри числа

    }

    cout << "Сума цифр числа: " << sum << endl; // виведення результату, використано оператор виведення

    return 0; // завершення головної ф-ції, повернення значення 0.

}

**4. Результати виконання завдань, тестування та фактично затрачений час:**

Завдання №1 VNS Practice Work 1.



Рисунок Результат виконання програми 1

Завдання №2 VNS Practice Work 2.

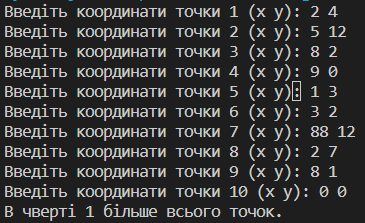


Рисунок Результат виконання програми 2

Завдання №3 VNS Practice Work 3.



Рисунок Результат виконання програми 3

Завдання №4 VNS Practice Work 4.



Рисунок Результат виконання програми 4

**Контрольні запитання**

1. **Назвіть основні властивості алгоритму.** Основні властивості алгоритму - дискретність, визначеність, виконуваність, скінченність, результативність, масовість, ефективність.
2. **Що таке алгоритм?** Алгоритм – це точна послідовність інструкцій для вирішення конкретної задачі або класу задач.
3. **Визначте основні етапи розробки алгоритмів.** Основні етапи розробки - постановка задачі, побудова математичної моделі, розробка алгоритму, опис алгоритму мовою програмування, тестування та налагоджування програм, експлуатація програми.
4. **Перелічить базові конструкції.** Послідовність, розгалуження, цикл.
5. **Перелічить складні базові конструкції.** Вкладені розгалуження, вкладені цикли.

1. **Дайте визначення конструкції розгалуження.** Конструкція розгалуження використовується для вибору між двома або більше альтернативами залежно від умови.
2. **Дайте визначення конструкції цикл.** Конструкція циклу дозволяє виконувати певні інструкції або блок коду декілька разів.
3. **Сформулюйте правило виконання циклу з передумовою.** Цикл виконується доти, доки виконується певна умова.
4. **Сформулюйте правило виконання циклу з відомою кількістю повторювань тіла циклу.** Цикл виконується заздалегідь визначену кількість разів.
5. **Що таке обчислювальна складність алгоритму?** Обчислювальна складність визначає кількість ресурсів (часу та пам'яті), які потрібні для виконання алгоритму.
6. **Як оцінити обчислювальну складність?** Обчислювальну складність алгоритмів оцінюють за кількістю дій, які виконує алгоритм та за обсягом задіяної пам'яті.
7. **Рекурсивні функції. Переваги їх використання.** Рекурсивні функції викликають самі себе для вирішення підзадач, зручні, роблять код читабельним, але мають обмеження на швидкодію для деяких завдань.

**Висновки:** В результаті виконання розрахункової роботи я закріпила навички програмування, які ми отримали протягом семестру. Повторила базові математичні задачі за допомогою програмування.