

Міністерство освіти і науки України
Національний університет «Львівська політехніка»
Кафедра систем штучного інтелекту



Звіт

**про виконання розрахунково-графічних робіт блоку № 7
з дисципліни: «Основи програмування»**

до:

ВНС Розрахунково-графічних робіт № 1-4
Практичних Робіт до блоку № 7

Виконав:

Студент групи ШІ-11
Фарина Арсеній Петрович

Львів 2024

Мета роботи:

Опанувати вивчений матеріал на практиці, а саме попрацювати з файлами як у мові C та C++, вдосконалити роботу зі стрічками в C++.

Теоретичні відомості:

- Теми пройдені під час семестру

Джерела використані для опрацювання теоретичного матеріалу:

- Ознайомився під час навчання

Виконання роботи:

1. Опрацювання завдання до програм.

Завдання №1

VNS PRACTICE WORK TASK 1 (VARIANT 24)

Розробити лінійний алгоритм для розв'язання задачі.

Варіант 24. $P = \sqrt[3]{m \cdot \operatorname{tg} t + |c \cdot \sin t|}$; $K = \cos^2 t - m / \sqrt[3]{|m^2 - t^2|}$, де $m=2,1$; $t=1,02$; $c=-1$.

Завдання №2

VNS PRACTICE WORK TASK (VARIANT 2)

Розробити алгоритм, що розгалужується.

$$\text{Варіант 2. } z = \begin{cases} a^x + b^y, & x + y \leq 1, \\ ax^2 + \ln(bxy), & x + y > 1; \end{cases}$$

де $x \in [1,2]$; $h_x = 0,1$; $a = 1$; $b = 2$; $y \in [1,2]$; $h_y = 0,2$.

Завдання №3

VNS PRACTICE WORK TASK 3 (VARIANT 18)

Перетворення введене з клавіатури дробове число в грошовий формат.

Наприклад, число 12,5 повинне бути перетворено до вигляду 12 грн. 50 коп.

Перетворення числа в грошовий формат.

Введіть дробове число > 23.6

23.6 грн. - це 23 грн. 60 коп.

Реалізувати виконання розрахунків: Верстат-автомат

виготовляє за 1 годину КОЛ шайб. Скільки шайб він виготовить за 8-и годинну зміну, за робочий день при двозмінній роботі і за 30-ти денний місяць, якщо в місяці 4 вихідних дня і в кожній зміні верстат знаходиться 1 годину на профілактиці?

Завдання №4

VNS PRACTICE WORK TASK 4 (VARIANT 17)

Скласти програму, яка генерує три послідовності з

десяти випадкових чисел в діапазоні від 1 до 10, виводить кожну

послідовність на екран і обчислює середнє арифметичне кожній послідовності.

Завдання №5

SELF PRACTICE WORK 1 ALGOTESTER (Числа для малят)

Малята, напишіть на клаптику паперу ціле додатне число n . Тепер уявіть собі, що Ви можете переставляти цифри у його десятковому записі як завгодно. Єдина умова — не повинно бути нулів на початку запису числа. Вам потрібно визначити мінімальне та максимальне числа, які можна отримати таким способом. Бажаю успіху! І не забувайте, що ви все знаєте, просто можливо щось забули.

Вхідні дані

У єдиному рядку задано одне ціле число n .

Вихідні дані

У єдиному рядку виведіть два цілих числа — мінімальне та максимальне числа, які можна отримати перестановкою цифр числа n .

Завдання №6

SELF PRACTICE WORK 2 ALGOTESTER (Зуби)

Мале Бісеня любить гострити зуби. А Зла Тітонька любить до нього підходити і питатися: «Що, зуби гостриш?». Бісеняті таке не дуже подобається, тому воно придумало робити таке.

У Малого Бісеняти є nn зубів. Кожен зуб має коефіцієнт загостреності $aiai$. Також існує межа загостреності kk . Якщо коефіцієнт загостреності певного зуба є більшим чи рівним межі загостреності, то такий зуб вважається загостреним. Мале Бісеня хоче наступного разу, коли Зла Тітонька його щось запитає, показати їй якнайбільше загострених зубів, що розташовані поспіль.

Допоможіть Малому Бісеняті дізнатися, скільки найбільше зубів воно зможе показати.

Вхідні дані

У першому рядку задані два цілих числа nn та kk — кількість зубів та межа загостреності відповідно.

В другому рядку задано nn цілих чисел $aiai$ — коефіцієнти загостреності зубів.

Вихідні дані

Єдине ціле число — відповідь на задачу.

Завдання №7

SELF PRACTICE WORK 3 ALGOTESTER (Офісна вулиця частина 1)

Зустрілися якось працівники великих компаній і почали... Обговорювати план вулиці.

Виявляється, всі приміщення, які орендуватимуть ці компанії, збудують вздовж однієї вулиці.

ii-та компанія орендуватиме офіс довжиною l_i метрів. Офіси будуватимуть один за одним, починаючи з точки 0. Всі працівники приїжджатимуть на стоянку, яку побудують в точці 0, та будуть йти до офісів своїх компаній.

Тобто, якщо офіси будуть збудовані в порядку p_1, p_2, \dots, p_n , то перший офіс почнеться в точці 0 і закінчиться в точці l_{p_1} , другий почнеться в l_{p_1} і закінчиться в $l_{p_1} + l_{p_2}$ і т.д. Двері кожного офісу завжди є в кінці будинку, який є ближчим до стоянки.

Ваше завдання — допомогти розмістити офіси компаній на цій вулиці в такому порядку, щоб сумарна відстань від точки 0 до усіх офісів була мінімальною.

Вхідні дані

У першому рядку задане ціле число n — кількість компаній.

У наступному рядку задано n цілих чисел l_i через пробіл — довжини офісів усіх компаній.

Вихідні дані

У єдиному рядку виведіть n чисел від 1 до n — порядок компаній, в якому варто будувати офіси.

Якщо існує декілька оптимальних порядків — виведіть будь-який із них.

Завдання №8

SELF PRACTICE WORK 4 ALGOTESTER (ALGOTESTER LAB 1v3)

Персонажу по одному дають сторони 5 кубів a_i , з яких він будує піраміду. Коли він отримує куб з ребром a_i - він його ставить на існуючий, перший ставить на підлогу (вона безмежна). Якщо в якийсь момент об'єм куба у руці (який будуть ставити) буде більший ніж у куба на вершині піраміди - персонаж програє і гра закінчується. Розмір усіх наступних кубів після програшу не враховується. Тобто якщо $a_{i-1} < a_i$ - це програш.

Ваше завдання - сказати як закінчиться гра.

2. Дизайн та планувальна оцінка часу виконання завдань:

Програма №1

- Блок-схема

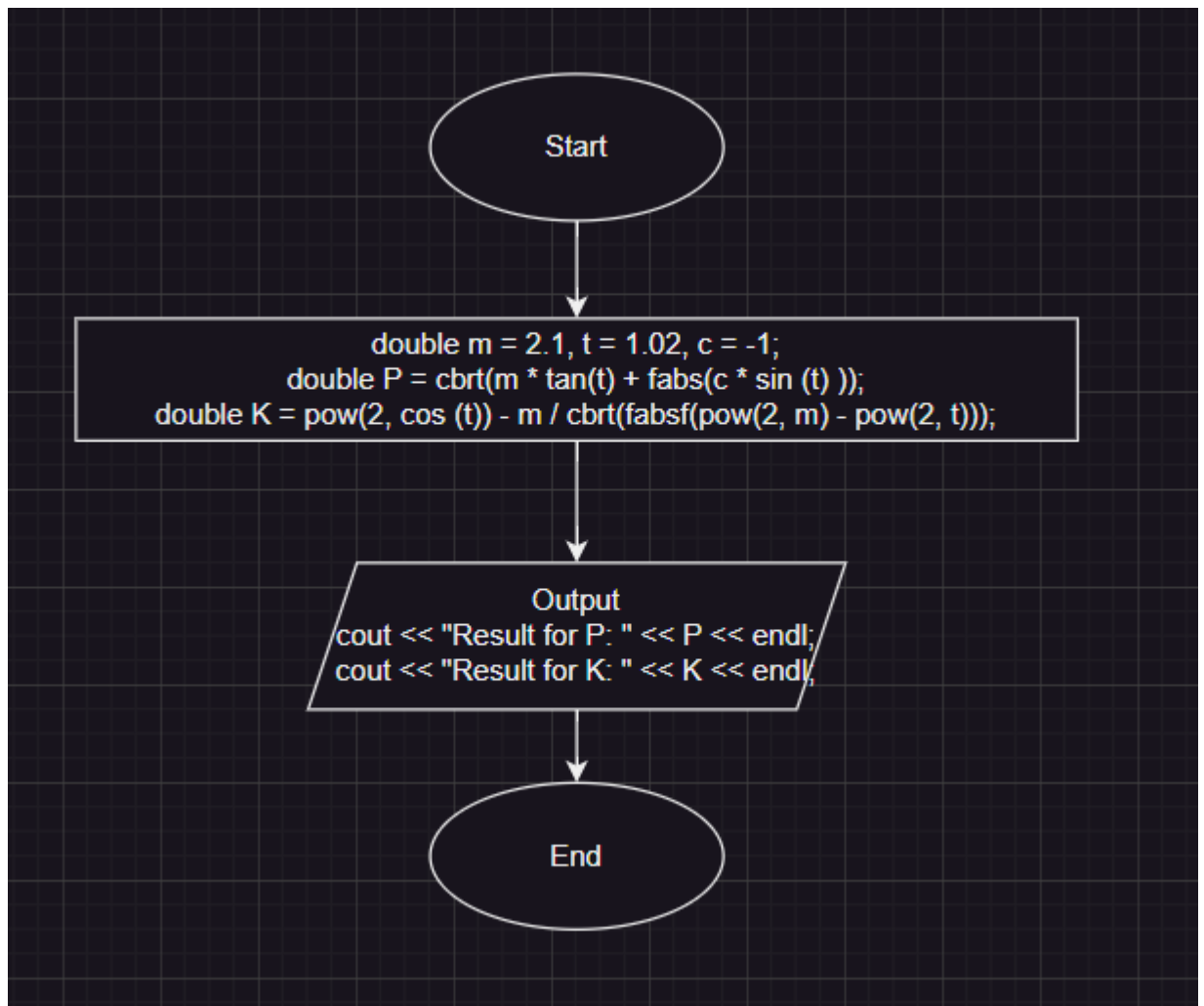


Рисунок 2.1. Блок-схема до програми №1

- Важливі деталі для реалізації програми
Використовувати тип `double` для точнішого результату, а також функцію `pow` з бібліотеки `cmath`.
- Плановий час на реалізацію 20 хвилин.

Програма №2

- Блок-схема

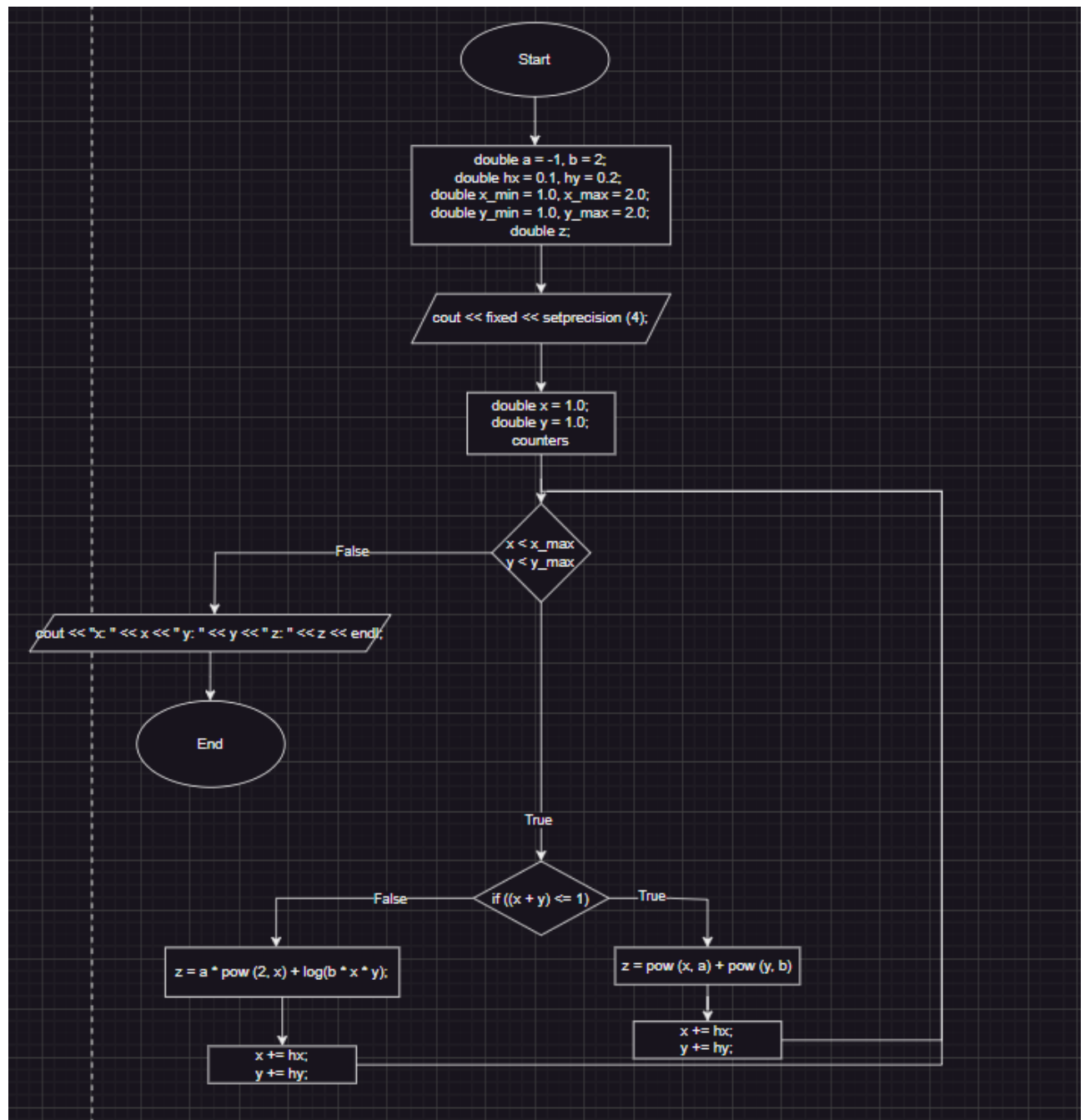


Рисунок 2.2. Блок-схема до програми №2

- Важливі деталі для реалізації програми
Використовувати цикли та умовні оператори для реалізації програми, а також тип double для точнішого результату.
- Плановий час на реалізацію 30 хвилин.

Програма №3

- Блок-схема

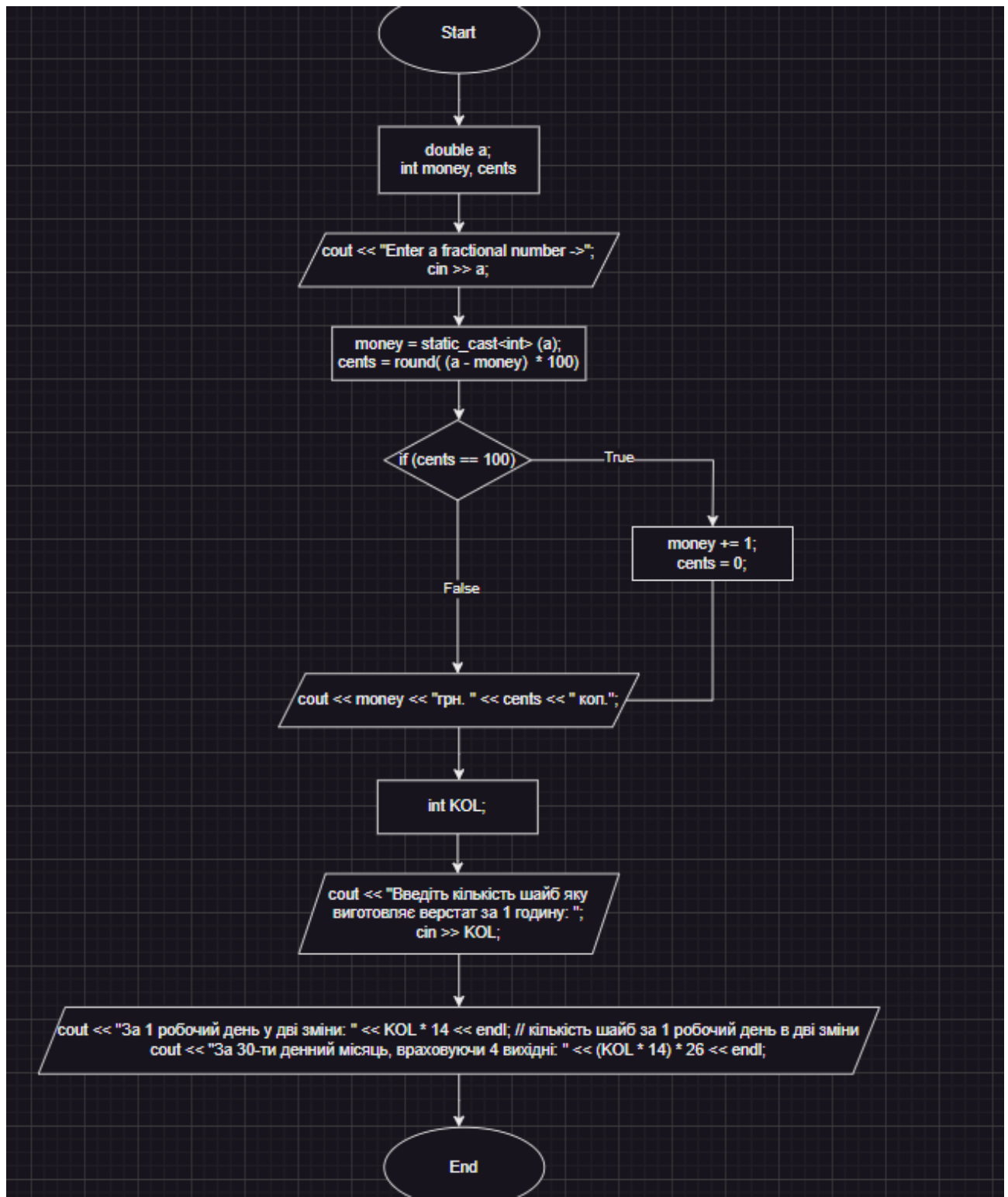
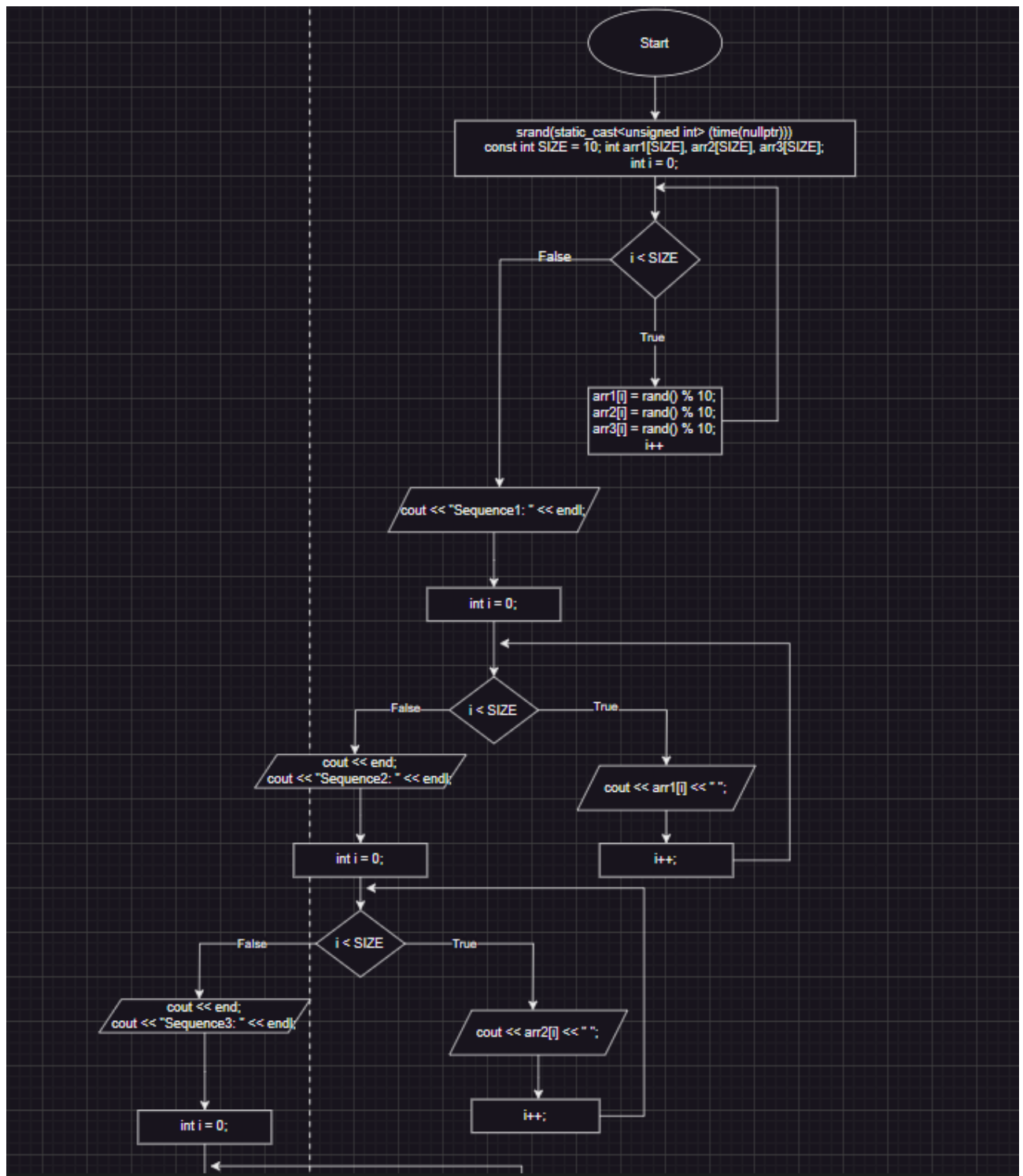


Рисунок 2.3. Блок-схема до програми №3

- Важливі деталі для реалізації програми
Використовувати функції бібліотеки `cmath`, а також умовний оператор для врахування деяких умов.
- Плановий час на реалізацію 45 хвилин.

Програма №4

- Блок-схема



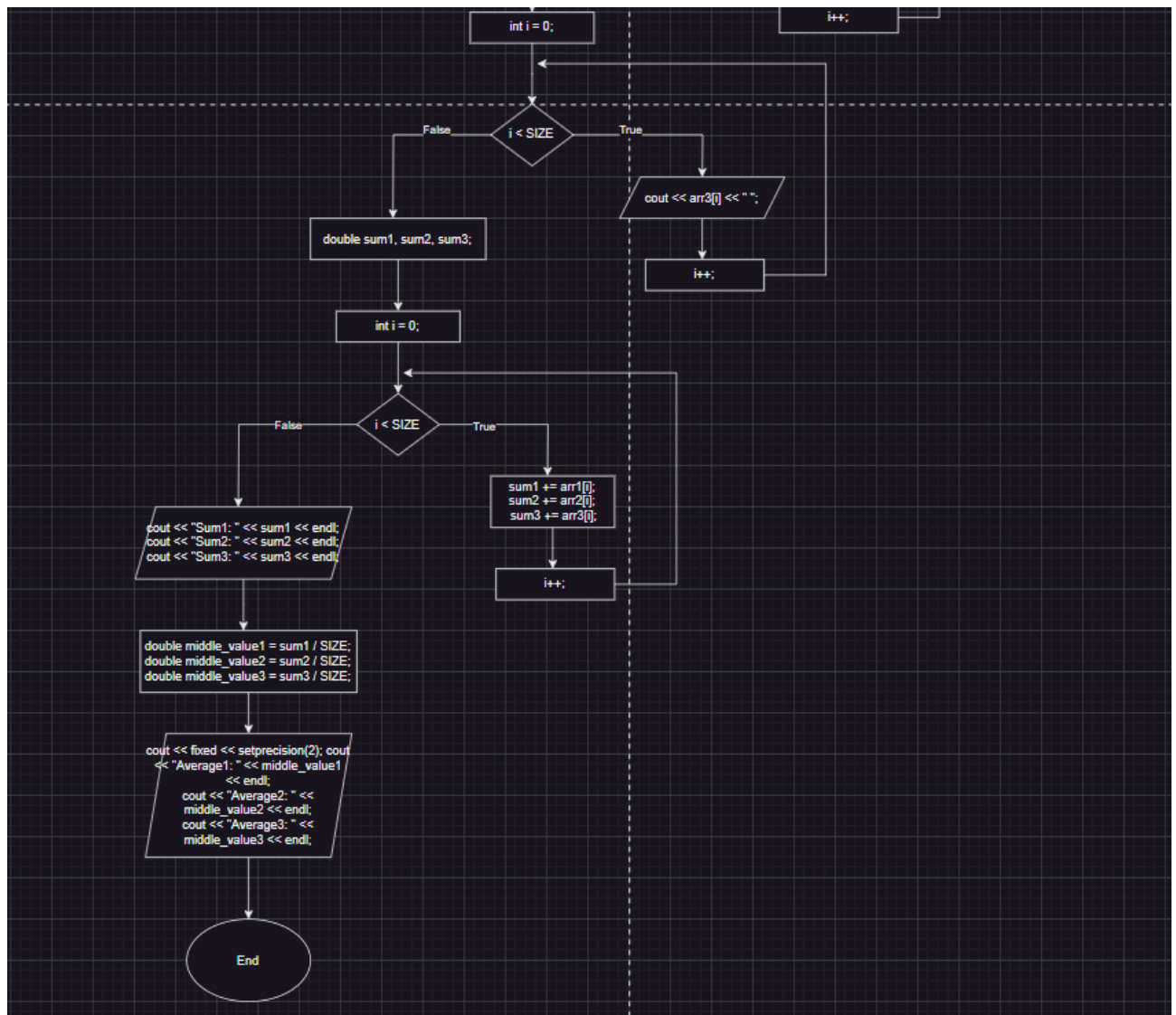


Рисунок 2.4. Блок-схема до програми №4

- Важливі деталі для реалізації програми

Для генерації випадкових чисел використовувати `rand` та `srand` (для різноманітності послідовностей випадкових чисел) та бібліотеку `ctime` для коректної роботи функції `srand`.

- Плановий час на реалізацію 50 хвилин.

Програма №5

- Блок-схема

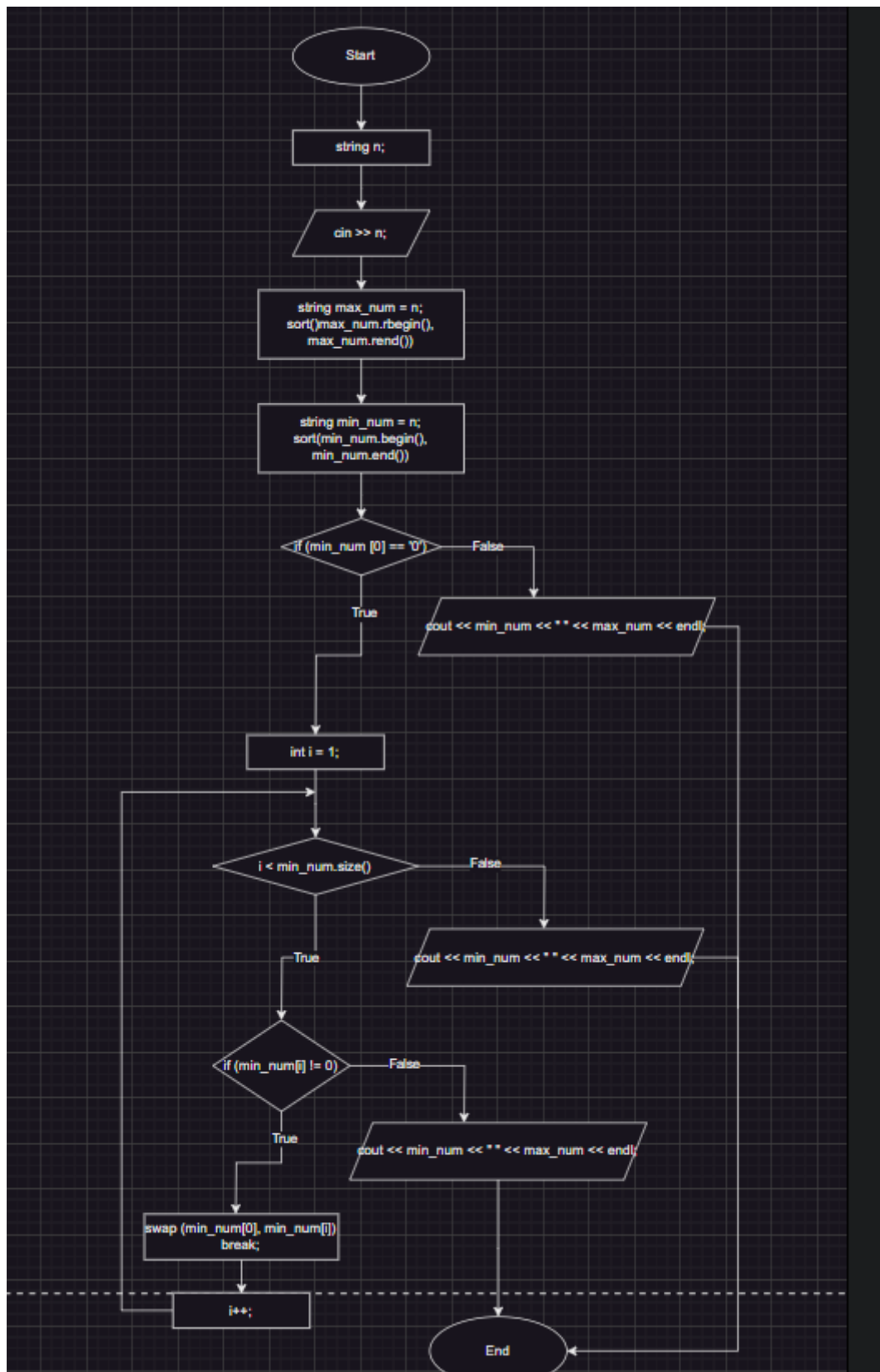


Рисунок 2.5. Блок-схема до програми №5

- Важливі деталі для реалізації програми

Використовувати тип даних `long long` через відповідний розмір вхідних даних.

- Плановий час на реалізацію 30 хвилин.

Програма №6

- Блок-схема

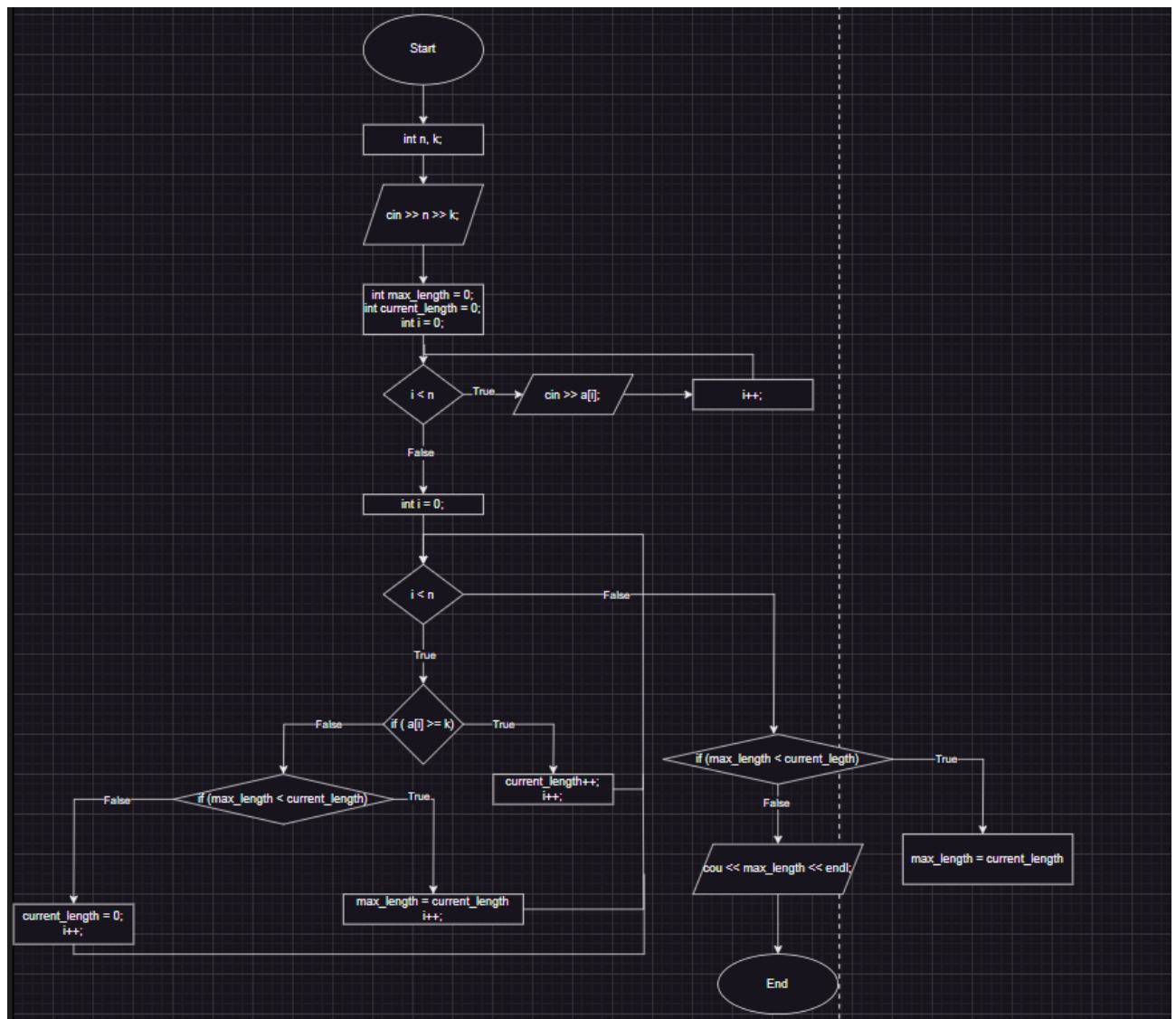


Рисунок 2.6. Блок-схема до програми №6

- Важливі деталі для реалізації програми
- Використовуючи вектор записати “рівень загострення зуба”, а потім за допомогою циклу реалізувати знаходження в масиві елементів, які більші або рівні за коефіцієнт загострення.
- Плановий час на реалізацію 30 хвилин.

Програма №7

- Блок-схема

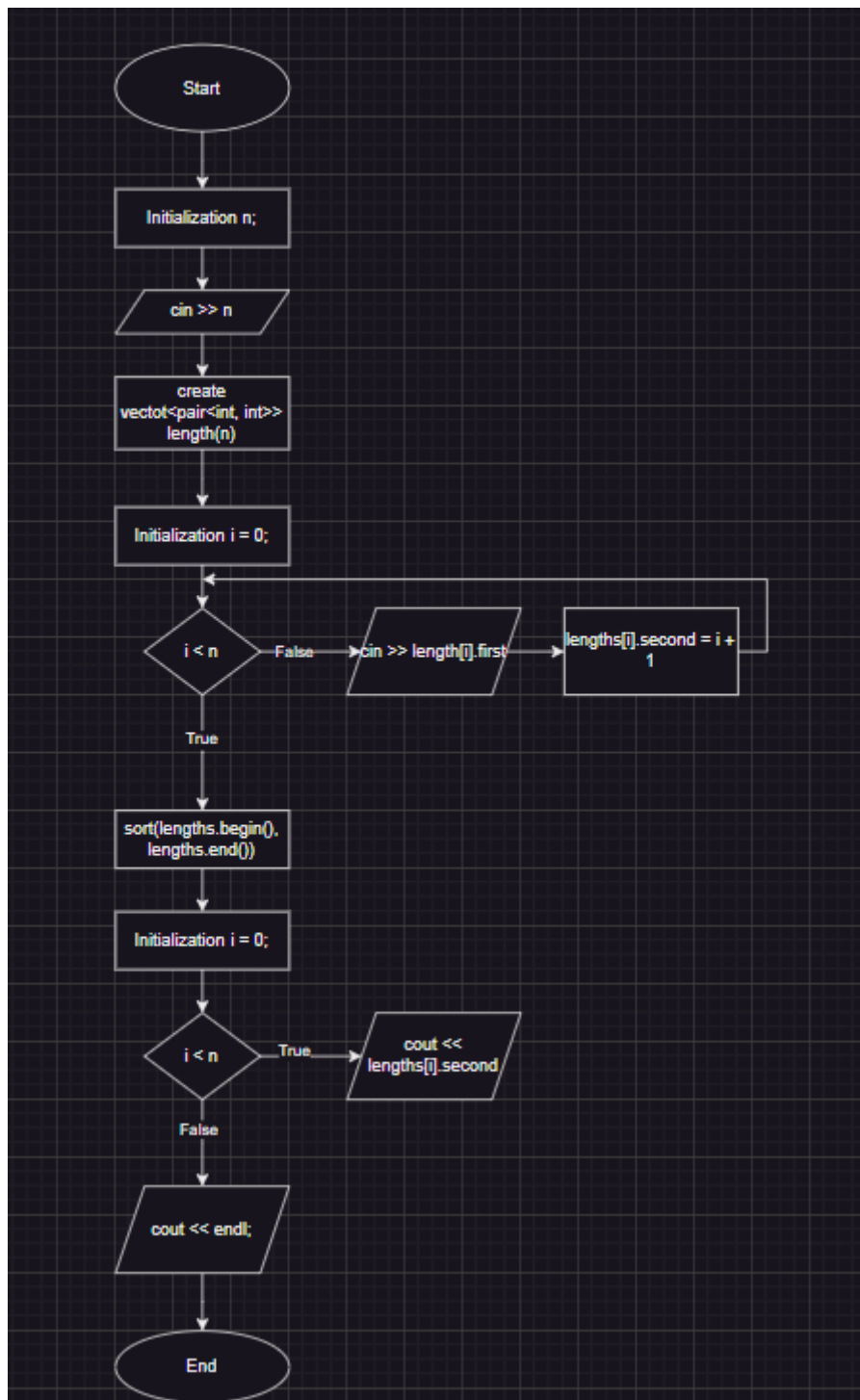


Рисунок 2.7. Блок-схема до програми 7

- Важливі деталі для реалізації програми
- Використати цикл для перевірки елементів у векторі.
- Плановий час на реалізацію 45 хвилин.

Програма №8

- Блок-схема

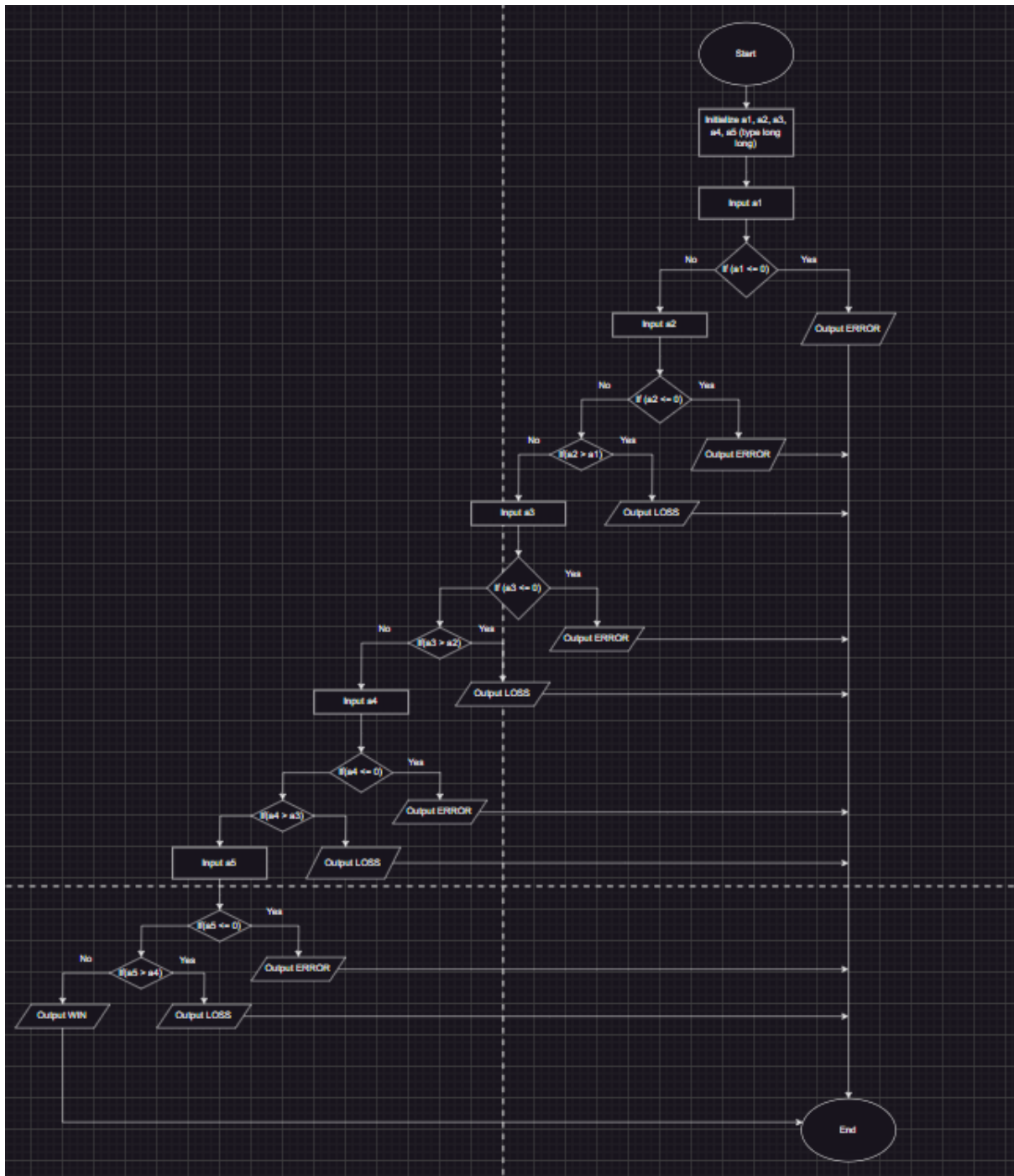



Рисунок 2.8. Блок-схема до програми 8

- Важливі деталі для реалізації програми
- Використовувати умовні оператори для реалізації задачі

Плановий час на реалізацію 30 хвилин.

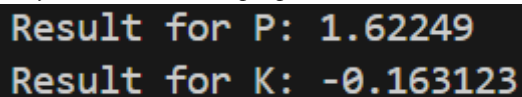
3. Код програм з посиланням на зовнішні ресурси та фактично затрачений час:

Завдання №1



```
1  #include <iostream>
2  #include <cmath>
3  #include <iomanip>
4
5  using namespace std;
6
7  int main(){
8
9      double m = 2.1, t = 1.02, c = -1;
10
11     double P = cbrt(m * tan(t) + fabs(c * sin (t) ));
12     double K = pow(2, cos (t)) - m / cbrt(fabsf(pow(2, m) - pow(2, t)));
13
14     cout << "Result for P: " << P << endl;
15     cout << "Result for K: " << K << endl;
16
17     return 0;
18 }
19
20
21
22
```

Рисунок 3.1. Код до програми № 1



```
Result for P: 1.62249
Result for K: -0.163123
```


Рисунок 3.2. Приклад виконання програми № 1

Фактично затрачений час 30 хвилин.

Посилання на файл у пулл реквесті

https://github.com/artificial-intelligence-department/ai_programming_playground_2024/pull/381

Завдання №2



```
1  #include <iostream>
2  #include <iomanip>
3  #include <cmath>
4
5  using namespace std;
6
7  int main (){
8
9      double a = 1, b = 2; // fixed value
10     double hx = 0.1, hy = 0.2; // steps
11
12     double x_min = 1.0, x_max = 2.0;
13     double y_min = 1.0, y_max = 2.0;
14     double z; // for result
15
16     cout << fixed << setprecision(4);
17
18     for (double x = 1.0; x < x_max; x += hx){
19         for (double y = 1.0; y < y_max; y += hy){
20             if ( (x + y) <= 1){
21                 z = pow (x, a) + pow (y, b);
22                 cout << "x: " << x << " y: " << y << " z: " << z << endl;
23             }
24             else {
25                 z = a * pow (2, x) + log(b * x * y);
26                 cout << " x: " << x << " y: " << y << " z: " << z << endl;
27             }
28         }
29     }
30
31     return 0;
32 }
```

Рисунок 3.3. Код до програми № 2

```
x: 1.0000 y: 1.0000 z: 2.6931
x: 1.0000 y: 1.2000 z: 2.8755
x: 1.0000 y: 1.4000 z: 3.0296
x: 1.0000 y: 1.6000 z: 3.1632
x: 1.0000 y: 1.8000 z: 3.2809
x: 1.0000 y: 2.0000 z: 3.3863
x: 1.1000 y: 1.0000 z: 2.9320
x: 1.1000 y: 1.2000 z: 3.1143
x: 1.1000 y: 1.4000 z: 3.2685
x: 1.1000 y: 1.6000 z: 3.4020
x: 1.1000 y: 1.8000 z: 3.5198
x: 1.1000 y: 2.0000 z: 3.6252
x: 1.2000 y: 1.0000 z: 3.1729
x: 1.2000 y: 1.2000 z: 3.3552
x: 1.2000 y: 1.4000 z: 3.5093
x: 1.2000 y: 1.6000 z: 3.6429
x: 1.2000 y: 1.8000 z: 3.7607
x: 1.2000 y: 2.0000 z: 3.8660
x: 1.3000 y: 1.0000 z: 3.4178
x: 1.3000 y: 1.2000 z: 3.6001
x: 1.3000 y: 1.4000 z: 3.7543
x: 1.3000 y: 1.6000 z: 3.8878
x: 1.3000 y: 1.8000 z: 4.0056
x: 1.3000 y: 2.0000 z: 4.1109
x: 1.4000 y: 1.0000 z: 3.6686
x: 1.4000 y: 1.2000 z: 3.8510
x: 1.4000 y: 1.4000 z: 4.0051
```



```
x: 1.5000 y: 1.4000 z: 4.2635
x: 1.5000 y: 1.6000 z: 4.3970
x: 1.5000 y: 1.8000 z: 4.5148
x: 1.5000 y: 2.0000 z: 4.6202
x: 1.6000 y: 1.0000 z: 4.1946
x: 1.6000 y: 1.2000 z: 4.3769
x: 1.6000 y: 1.4000 z: 4.5311
x: 1.6000 y: 1.6000 z: 4.6646
x: 1.6000 y: 1.8000 z: 4.7824
x: 1.6000 y: 2.0000 z: 4.8877
x: 1.7000 y: 1.0000 z: 4.4728
x: 1.7000 y: 1.2000 z: 4.6551
x: 1.7000 y: 1.4000 z: 4.8093
x: 1.7000 y: 1.6000 z: 4.9428
x: 1.7000 y: 1.8000 z: 5.0606
x: 1.7000 y: 2.0000 z: 5.1659
x: 1.8000 y: 1.0000 z: 4.7631
x: 1.8000 y: 1.2000 z: 4.9455
x: 1.8000 y: 1.4000 z: 5.0996
x: 1.8000 y: 1.6000 z: 5.2331
x: 1.8000 y: 1.8000 z: 5.3509
x: 1.8000 y: 2.0000 z: 5.4563
x: 1.9000 y: 1.0000 z: 5.0671
x: 1.9000 y: 1.2000 z: 5.2495
x: 1.9000 y: 1.4000 z: 5.4036
x: 1.9000 y: 1.6000 z: 5.5371
x: 1.9000 y: 1.8000 z: 5.6549
x: 1.9000 y: 2.0000 z: 5.7603
```

Рисунок 3.4. Приклад виконання програми № 2

Фактично затрачений час 40 хвилин.

Посилання на файл у пулл реквесті

https://github.com/artificial-intelligence-department/ai_programming_playground_2024/pull/381

Завдання №3

```

1  #include <iostream>
2  #include <cmath> // для round
3
4  using namespace std;
5
6  int main (){
7      cout << "--)-Money-(-----" << endl;
8      double a;
9      int money, cents;
10
11     cout << "Enter a fractional number -> ";
12     cin >> a;
13
14     money = static_cast<int> (a);
15     cents = round((a - money) * 100);
16
17     if (cents == 100){
18         money += 1;
19         cents = 0;
20     }
21
22     cout << money << "грн. " << cents << " коп.";
23
24     cout << "--)-Calculation_task-(-----" << endl;
25
26     int KOL; // кількість шайб яку виготовить верстат за 1 годину
27     cout << "Введіть кількість шайб яку виготовляє верстат за 1 годину: ";
28     cin >> KOL;
29
30     cout << "За 1 робочий день у дві зміни: " << KOL * 14 << endl; // кількість шайб за 1 робочий день в дві зміни
31     cout << "За 30-ти денний місяць, враховуючи 4 вихідні: " << (KOL * 14) * 26 << endl;
32
33     return 0;
34 }
35

```

Рисунок 3.5. Код до програми № 3

```

--)-Money-(-----
Enter a fractional number -> 12.5
12грн. 50 коп.--)-Calculation_task-(-----
Введіть кількість шайб яку виготовляє верстат за 1 годину: 4
За 1 робочий день у дві зміни: 56
За 30-ти денний місяць, враховуючи 4 вихідні: 1456
PS D:\ai_programming_playground_2024\ai_11\faryna_arsenii\Saga1>

```

Рисунок 3.6. Приклад виконання програми № 3

Фактично затрачений час 35 хвилин.

Посилання на файл у пулл реквесті

https://github.com/artificial-intelligence-department/ai_programming_playground_2024/pull/381

Завдання №4

```

1  #include <iostream>
2  #include <ctime>
3  #include <iomanip>
4
5  using namespace std;
6
7  int main (){
8
9      srand(static_cast<unsigned int> (time(nullptr)));
10
11     const int SIZE = 10;
12     int arr1[SIZE], arr2[SIZE], arr3[SIZE];
13
14     for (int i = 0; i < SIZE; i++){
15         arr1[i] = rand() % 10;
16         arr2[i] = rand() % 10;
17         arr3[i] = rand() % 10;
18     }
19
20     cout << "Sequence1: ";
21     for (int i = 0; i < SIZE; i++){
22         cout << arr1[i] << " ";
23     }
24     cout << endl;
25
26     cout << "Sequence2: ";
27     for (int i = 0; i < SIZE; i++){
28         cout << arr2[i] << " ";
29     }
30     cout << endl;
31
32     cout << "Sequence3: ";
33     for (int i = 0; i < SIZE; i++){
34         cout << arr3[i] << " ";
35     }
36     cout << endl;
37
38     double sum1, sum2, sum3;
39
40     for (int i = 0; i < SIZE; i++){
41         sum1 += arr1[i];
42         sum2 += arr2[i];
43         sum3 += arr3[i];
44     }
45
46     cout << "Sum1: " << sum1 << endl;
47     cout << "Sum2: " << sum2 << endl;
48     cout << "Sum3: " << sum3 << endl;
49
50     double middle_value1 = sum1 / SIZE;
51     double middle_value2 = sum2 / SIZE;
52     double middle_value3 = sum3 / SIZE;
53
54     cout << fixed << setprecision(2);
55     cout << "Average1: " << middle_value1 << endl;
56     cout << "Average2: " << middle_value2 << endl;
57     cout << "Average3: " << middle_value3 << endl;
58
59     return 0;
60 }

```

Рисунок 3.7. Код до програми № 4

```
Sequence1: 4 3 8 6 9 9 6 9 3 9
Sequence2: 9 6 5 4 8 9 7 5 8 3
Sequence3: 0 1 0 3 7 9 0 6 8 7
Sum1: 66
Sum2: 64
Sum3: 41
Average1: 6.60
Average2: 6.40
Average3: 4.10
```

Рисунок 3.8. Приклад виконання програми № 4

Фактично затрачений час 30 хвилин.

Посилання на файл у пулл реквесті

https://github.com/artificial-intelligence-department/ai_programming_playground_2024/pull/381

Завдання №5

```
1  #include <iostream>
2  #include <string>
3  #include <algorithm>
4
5  using namespace std;
6
7  int main (){
8
9      string n;
10     cin >> n;
11
12     string max_num = n;
13     sort(max_num.rbegin(), max_num.rend()); //за спаданням
14
15     string min_num = n;
16     sort(min_num.begin(), min_num.end()); //за зростанням
17
18     if (min_num[0] == '0'){
19         for (int i = 1; i < min_num.size(); i++){
20             if (min_num[i] != '0'){
21                 swap (min_num[0], min_num[i]);
22                 break;
23             }
24         }
25     }
26
27     cout << min_num << " " << max_num << endl;
28
29     return 0;
30 }
31
```

Рисунок 3.9. Код до програми № 5

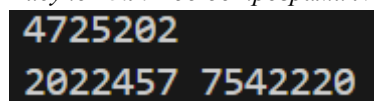


Рисунок 3.10. Приклад виконання програми № 5

Створено	Компілятор	Результат	Час (сек.)	Пам'ять (МіБ)	Дії
декілька секунд тому	C++ 23	Зараховано	0.002	1.156	Перегляд


Рисунок 3.11. Статус задачі на Algotester

Фактично затрачений час 45 хвилин.

Посилання на файл у пулл реквесті

https://github.com/artificial-intelligence-department/ai_programming_playground_2024/pull/381

Завдання №6

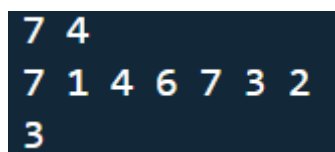


```

1  #include <iostream>
2  #include <vector>
3
4  using namespace std;
5
6  int main() {
7      int n, k;
8      cin >> n >> k;
9      vector<int> a(n);
10
11     int max_length = 0;
12     int current_length = 0;
13
14     for (int i = 0; i < n; i++) {
15         cin >> a[i];
16     }
17
18     for (int i = 0; i < n; i++) {
19         if (a[i] >= k) {
20             current_length++;
21         }
22         else {
23             if (max_length < current_length){
24                 max_length = current_length;
25             }
26             current_length = 0;
27         }
28     }
29     if (max_length < current_length){
30         max_length = current_length;
31     }
32
33     cout << max_length << endl;
34
35     return 0;
36 }
37

```

Рисунок 3.12. Код до програми № 6



```

7 4
7 1 4 6 7 3 2
3
3

```

Рисунок 3.13. Приклад виконання програми №6

Створено	Компілятор	Результат	Час (сек.)	Пам'ять (МіБ)	Дії
декілька секунд тому	C++ 23	Зараховано	0.044	1.469	Перегляд

Рисунок 3.14. Статус задачі на Algotester

Фактично затрачений час 30 хвилин.

Посилання на файл у пулл реквесті

https://github.com/artificial-intelligence-department/ai_programming_playground_2024/pull/381

Завдання №7

```
1  #include <iostream>
2  #include <vector>
3  #include <algorithm>
4
5  using namespace std;
6
7  int main() {
8      int n;
9      cin >> n; // Кількість компаній
10     vector<pair<int, int>> lengths(n); // Вектор пар для довжин і індексів
11
12     // Зчитування довжин офісів
13     for (int i = 0; i < n; i++) {
14         cin >> lengths[i].first; // Довжина офісу
15         lengths[i].second = i + 1; // Зберігаємо індекс (1-based)
16     }
17
18     // Сортують пар за довжинами
19     sort(lengths.begin(), lengths.end());
20
21     // Виведення порядку індексів
22     for (int i = 0; i < n; i++) {
23         cout << lengths[i].second << " "; // Виводимо індекси
24     }
25     cout << endl;
26
27     return 0;
28 }
```

Рисунок 3.15. Код до програми № 7

```
5
3 5 9 3 5
1 4 2 5 3
```

Рисунок 3.16. Приклад виконання програми №7

Створено	Компілятор	Результат	Час (сек.)	Пам'ять (МіБ)	Дії
17 днів тому	C++ 23	Зараховано	0.051	2.297	Перегляд

Рисунок 3.17. Статус завдання на Algotester

Фактично затрачений час 1 година.

Посилання на файл у пулл реквесті

https://github.com/artificial-intelligence-department/ai_programming_playground_2024/pull/381

Завдання №8

```
1  #include <iostream>
2
3  using namespace std;
4
5  int main() {
6      long long a1, a2, a3, a4, a5;
7      cin >> a1;
8      if (a1 <= 0) {
9          cout << "ERROR" << endl;
10         return 0;
11     }
12
13     cin >> a2;
14     if (a2 <= 0) {
15         cout << "ERROR" << endl;
16         return 0;
17     }
18     if(a2>a1){
19         cout<<"LOSS";
20         return 0;
21     }
22
23     cin >> a3;
24     if (a3 <= 0) {
25         cout << "ERROR" << endl;
26         return 0;
27     }
28     if(a3>a2){
29         cout<<"LOSS";
30         return 0;
31     }
32
33     cin >> a4;
34     if (a4 <= 0) {
35         cout << "ERROR" << endl;
36         return 0;
37     }
38     if(a4>a3){
39         cout<<"LOSS";
40         return 0;
41     }
42
43     cin >> a5;
44     if (a5 <= 0) {
45         cout << "ERROR" << endl;
46         return 0;
47     }
48     if(a5>a4){
49         cout<<"LOSS";
50         return 0;
51     }
52     cout << "WIN";
53     return 0;
54 }
```

Рисунок 3.18. Код до програми № 8

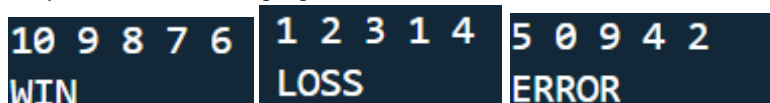


Рисунок 3.19. Приклад виконання програми №8

a few seconds ago	C++ 23	Accepted	0.003	1.238	View
-------------------	--------	----------	-------	-------	----------------------

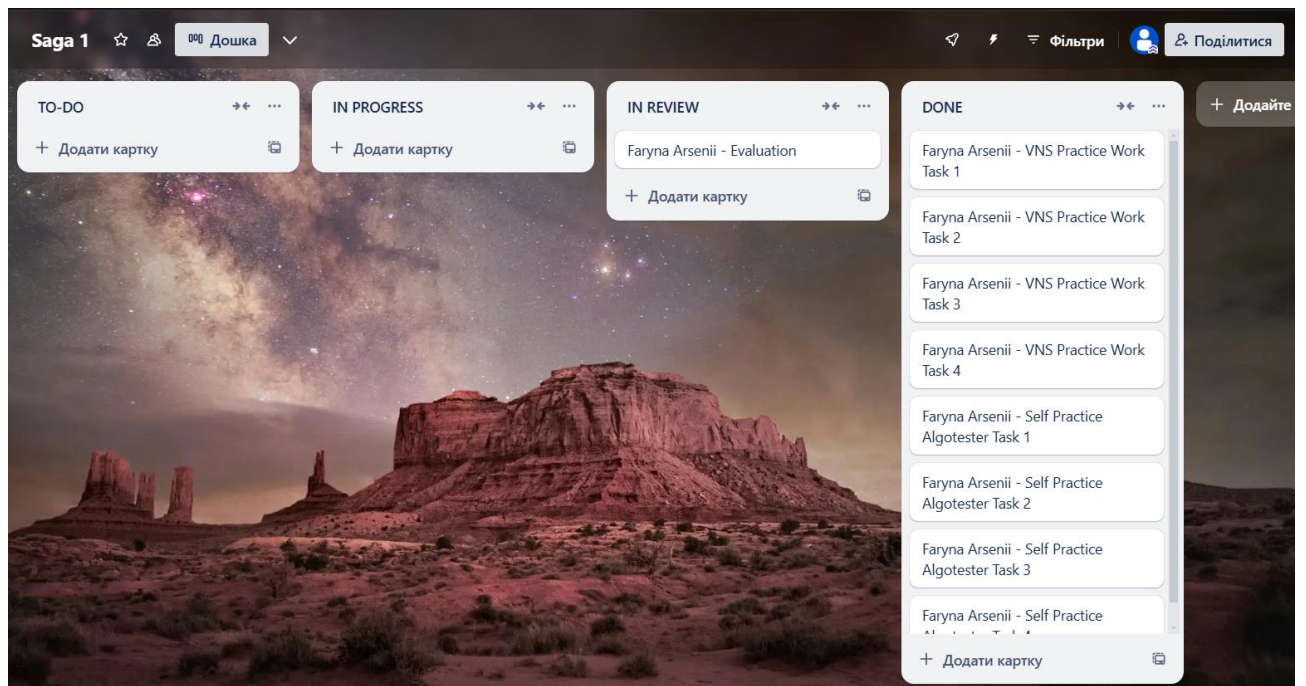
Рисунок 3.20. Статус задачі на Algotester

Фактично затрачений час 45 хвилин.

Посилання на файл у пулл реквесті

https://github.com/artificial-intelligence-department/ai_programming_playground_2024/pull/381

4. Робота з командою:



Висновок: виконавши цю розрахункову роботу я вдосконалював свої знання з мови програмування C++.