

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



# **Звіт**

**ДО РОЗРАХУНКОВОЇ РОБОТИ №1  
З КУРСУ ОСНОВИ ПРОГРАМУВАННЯ.**

**Виконав:**  
Студент групи ШІ-11  
Бубельник Юрій Олегович

Львів 2024

## Мета роботи:

Одержати практичні навички в розробці і дослідженні алгоритмів розв'язання задач.

## Теоретичні відомості:

## Виконання роботи:

### Завдання №1 – VNS Practice Work Task 1 v25

#### Задача:

Розробити лінійний алгоритм для розв'язання задачі.

$$\text{Варіант 25. } a = y + \frac{x}{y^3 + \left| \frac{x^2}{y + \sqrt[3]{x^2}} \right|}; \quad b = 1 + \operatorname{tg}^2(x/2), \text{ де } x=1,23;$$

$$y=0,79.$$

### Завдання №2 – VNS Practice Work Task 2 v1

#### Задача:

Розробити алгоритм, що розгалужується для розв'язання задачі номер якої відповідає порядковому номеру студента в журналі викладача

$$\text{Варіант 1. } y = \begin{cases} ax + b \cos x, & x < 0,5, \\ bx^2 + c \sin 2x, & 0,5 \leq x < 1; \end{cases}$$

$$\text{де } x \in [0,2]; h_x = 0,1; a = 0,75; b = 1,19; c = -2,5.$$

### Завдання №3 – VNS Practice Work Task 3 v19

**Варіант 19.** Обчислює площу трикутника, якщо відомі координати його кутів. Нижче приведений вид екрану під час виконання програми, що рекомендується (дані, введені користувачем, які вводяться напівжирним шрифтом).

Обчислення площі трикутника.

Введіть координати кутів

(числа розділяйте пропуском):

x1,y1 > **-2 5**

x2,y2 > **1 7**

x3,y3 > **5 -3**

Площа трикутника: 23.56 кв.см.

Обчислення вартості покупки, що складається з набору зошитів і олівців.

Додатково реалізував запис даних у файл та вивід цих даних у консоль, за вибором користувача.

### **Завдання №4 – VNS Practice Work Task 4 v16**

**Варіант 16.** Скласти програму, яка генерує послідовності з 10 випадкових чисел в діапазоні від 1 до 10, виводить ці числа на екран і обчислює їх середнє арифметичне.

## **Algotester Practice Work**

### **Завдання №5 Algotester Lab 1v1**

#### **Задача:**

У вашого персонажа є  $H$  хітпойнтів та  $M$  мани.

Персонаж 3 рази використає закляття, кожне з яких може використати хітпойнти та ману одночасно.

Якщо якесь закляття забирає і хітпойнти і ману - ваш персонаж програє, отже для виграшу треба використовувати при одному заклинанні АБО хітпойнти, АБО ману.

Якщо в кінці персонаж буде мати додатню кількість хітпойнтів та мани ( $H, M > 0$ ) - він виграє, в іншому випадку програє.

Ваше завдання у випадку виграшу персонажа вивести YES, вивести NO у іншому випадку.

## Input

2 цілих числа  $H$  та  $M$  - хітпойнти та мана персонажа

3 рядки по 2 цілих числа,  $h_i$  та  $m_i$  - кількість хітпойнтів та мани, які ваш персонаж потратить за хід на  $i$  заклинання

## Output

*YES* - якщо ваш персонаж виграє

*NO* - у всіх інших випадках

## Constraints

$$1 \leq H \leq 10^{12}$$

$$1 \leq M \leq 10^{12}$$

$$0 \leq h_i \leq 10^{12}$$

$$0 \leq m_i \leq 10^{12}$$

## Завдання №6 Algotester Lab 2v1

### Задача:

У вас є дорога, яка виглядає як NN чисел.

Після того як ви по ній пройдете - вашу втому можна визначити як різницю максимального та мінімального елементу.

Ви хочете мінімізувати втому, але все що ви можете зробити - викинути одне число з дороги, тобто забрати його з масиву.

В результаті цієї дії, яку мінімальну втому ви можете отримати в кінці дороги?

## Input

У першому рядку ціле число  $N$  - кількість чисел

У другому рядку масив  $r$ , який складається з  $N$  цілих чисел

## Output

Єдине ціле число  $m$  - мінімальна втома, яку можна отримати

## Constraints

$$1 \leq N \leq 10^5$$

$$0 \leq r_i \leq 10^5$$

## Завдання №7 Апельсини

### Задача:

Діти міряються різними речима. У нашій задачі — апельсинами.

Марічка і Софійка міряються з Петриком.

Потрібно визначити, чи Марічка й Софійка разом мають більше апельсинів, ніж Петрик.

### Input

У першому і єдиному рядку задано три цілі числа  $a$ ,  $b$  та  $c$  — кількість апельсинів у Марічки, Софійки і Петрика відповідно.

### Output

У єдиному рядку виведіть відповідь до задачі — **YES**, якщо дівчата разом мають більше апельсинів, ніж Петрик, або **NO** в іншому випадку.

### Constraints

$$1 \leq a, b, c \leq 10^9.$$

## Завдання №8 Algotester Lab 3v3

### Задача:

**Вам дана стрічка  $s$ .**

Ваше завдання зробити компресію стрічки, тобто якщо якась буква йде більше одного разу підряд у стрічці замінити її на букву + кількість входжень підряд.

## Input

У першому рядку стрічка  $S$

## Output

Стрічка  $S_{compressed}$

## Constraints

$$1 \leq |S| \leq 10^5$$

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

### Завдання №1:

#### VNS Practice Work Task 1 v25

```
vns_practice_work_1_task_yurii_bubelnyk.cpp > main()
1  #include <iostream>
2  #include <cmath> // підключив бібліотеку для pow() та tan()
3  using namespace std;
4
5  int main(){
6
7      double x = 1.23; // ініціалізація x
8      double y = 0.79; // ініціалізація y
9
10     double a = y + x / (pow(y, 3) + abs(pow(x, 2)/(y + pow(x, 2.0/3.0)))); // формула для a
11     double tg = tan( x / 2 ); // обрахунок тангенса
12     double b = 1 + pow(tg, 2); // формула для b
13
14     cout << "Value of a = " << a << "\t\tValue of b = " << b << endl << endl; // вивід значень
15     return 0;
16 }
```

### Завдання №2:

#### VNS Practice Work Task 2 v1

```

vns_practice_work_2_task_yuri_bubeinyk.cpp > calculate(double, double, double, double, double, double)
1  #include <iostream>
2  #include <cmath> // для cos() і sin()
3
4  using namespace std;
5
6  // Рекурсивна функція
7  void calculate(double x, double x_end, double hx, double a, double b, double c);
8
9  int main() {
10     double a = 0.75;
11     double b = 1.19;
12     double c = -2.5;
13     // Значення для X
14     double x_start = 0.0;
15     double x_end = 2.0;
16     double hx = 0.1;
17
18     // Виклик рекурсивної функції
19     calculate(x_start, x_end, hx, a, b, c);
20
21     return 0;
22 }
23
24 void calculate(double x, double x_end, double hx, double a, double b, double c) {
25     if (x > x_end) { // коли X вийде за межі тоді закінчиться функція
26         return;
27     }
28
29     double y; // Змінна для збереження обчислень
30     //Перевірка меж
31     if (x < 0.5) {
32         y = a * x + b; // Випадок для x < 0.5
33     } else if (x >= 0.5 && x < 1.0) {
34         y = cos(b * x) + c; // Випадок для 0.5 <= x < 1.0
35     } else {
36         y = sin(2 * x) + b * c; // Випадок для 1.0 <= x <= 2.0
37     }
38
39     // Вивід результатів
40     cout << "x = " << x << ", y = " << y << endl;
41
42     // Рекурсія, функція викликається з наступним кроком X
43     calculate(x + hx, x_end, hx, a, b, c);
44 }

```

### Завдання №3:

### VNS Practice Work Task 3 v19



```

vns_practice_work_2_task_yurii_bubelnik.cpp × vns_practice_work_3_task_yurii_bubelnik.cpp × Work3.txt
vns_practice_work_3_task_yurii_bubelnik.cpp > main()
1 #include <iostream>
2 #include <cmath> // підключення бібліотеки для abs()
3 #include <iomanip> // Для встановлення точності виведення
4 #include <cstdio> // Для fopen, fprintf, fclose
5
6 using namespace std;
7
8 const int BUFFER = 256;
9
10 double Square(double x1, double y1, double x2, double y2, double x3, double y3);
11
12 bool IsValid(double x1, double y1, double x2, double y2, double x3, double y3);
13
14 double calculateValue(int numNotebooks, int numPencils, double priceNotebook, double pricePencil);
15
16 void CreateFile(const char* filename, double (*Func)(int, int, double, double), int numNotebooks, int numPencils, double priceNotebook, double pricePencil);
17
18 void ShowFile(const char* filename);
19
20 int main() {
21     const char* filename = "Work3.txt";
22     double x1, y1, x2, y2, x3, y3 = 0;
23
24     do {
25         cout << "x1,y1 > " << endl;
26         cin >> x1 >> y1;
27         cout << "x2,y2 > " << endl;
28         cin >> x2 >> y2;
29         cout << "x3,y3 > " << endl;
30         cin >> x3 >> y3;
31
32         // Перевіряємо, чи введено правильні координати
33         if (IsValid(x1, y1, x2, y2, x3, y3)) {
34             break; // Виходимо з циклу, якщо координати правильні
35         } else {
36             cout << "Invalid coordinates! Points cannot be the same or form a degenerate triangle. Please try again." << endl;
37         }
38     } while (true); // Повторяється, поки користувач не введе коректні координати
39
40     // Встановлюємо точність до двох знаків після коми
41     cout << fixed << setprecision(2);
42     cout << "Area of the triangle: " << Square(x1, y1, x2, y2, x3, y3) << " sq.cm." << endl;
43
44     // Введення даних для покупки
45     int numNotebooks, numPencils;
46     double priceNotebook, pricePencil;
47
48     cout << "\nEnter number of notebooks: ";
49     cin >> numNotebooks;
50     cout << "Enter price per notebook: ";
51     cin >> priceNotebook;
52
53     cout << "Enter number of pencils: ";
54     cin >> numPencils;
55     cout << "Enter price per pencil: ";
56     cin >> pricePencil;
57
58

```

```

58     cout << "Enter price per pencil: ";
59     cin >> pricePencil;
60
61     cout << "Total cost of purchase: " << calculateValue(numNotebooks, numPencils, priceNotebook, pricePencil) << " UAH.\n";
62     int choice; // для запису вибору
63     // options для користувача (-_0)
64     do {
65         cout << "Enter choice:\n\t 1) Create the file;\n\t 2) Show the file;\n\t 3) Exit.\n";
66         cin >> choice;
67         switch (choice)
68         {
69             case 1:
70                 CreateFile(filename, calculateValue, numNotebooks, numPencils, priceNotebook, pricePencil);
71                 break;
72             case 2:
73                 ShowFile(filename);
74                 break;
75             case 3:
76                 cout << "Exit -_" << endl;
77             default:
78                 break;
79         }
80     } while (choice != 3);
81
82     return 0;
83
84 double Square(double x1, double y1, double x2, double y2, double x3, double y3) {
85     // Обчислення площі трикутника через координати
86     double square = 0;
87     // обчислюємо площу трикутника за допомогою координат за формулою площі через координати вершин.
88     square = 0.5 * ((x1 * y2 + x2 * y3 + x3 * y1) - (y1 * x2 + y2 * x3 + y3 * x1));
89     return abs(square); // Повертаємо абсолютне значення площі
90 }
91
92 bool IsValid(double x1, double y1, double x2, double y2, double x3, double y3) {
93     // Перевірка, чи всі точки різні, щоб утворити трикутник
94     if ((x1 == x2 && y1 == y2) || (x2 == x3 && y2 == y3) || (x1 == x3 && y1 == y3)) {
95         return false; // Точки не можуть бути однаковими
96     }
97     // Перевірка на вироджений трикутник (сума двох сторін повинна бути більшою за третю)
98     double side1 = sqrt((x2 - x1) * (x2 - x1) + (y2 - y1) * (y2 - y1));
99     double side2 = sqrt((x3 - x2) * (x3 - x2) + (y3 - y2) * (y3 - y2));
100    double side3 = sqrt((x3 - x1) * (x3 - x1) + (y3 - y1) * (y3 - y1));
101
102    if (side1 + side2 <= side3 || side2 + side3 <= side1 || side3 + side1 <= side2) {
103        return false; // Якщо три сторони не можуть утворити трикутник
104    }
105
106    return true; // Координати правильні для утворення трикутника
107 }
108
109 double calculateValue(int numNotebooks, int numPencils, double priceNotebook, double pricePencil) {
110     double sum = numNotebooks * priceNotebook + numPencils * pricePencil;
111     return sum;
112 }

```



```

// передаємо у функцію CreateFile функція calculateValue як pointer
void CreateFile(const char* filename, double (*Func)(int, int, double, double), int numNotebooks, int numPencils, double priceNotebook, double pricePencil){
    FILE* file = fopen(filename, "w"); // Відкриваємо файл для запису
    if (file == nullptr) {
        cerr << "Error while opening th file: " << filename << endl;
        return;
    }

    int result = Func(numNotebooks, numPencils, priceNotebook, pricePencil);

    // Записуємо результат у файл
    fprintf(file, "The total value is: %d\n", result);

    fclose(file); // Закриваємо файл
    cout << "Result has been written to the file: " << filename << endl;
}

// вивід даних з файлу в термінал
void ShowFile(const char* filename){
    FILE* file = fopen(filename, "r"); // Відкриваємо файл для запису
    if (file == nullptr) {
        cerr << "Error while opening th file: " << filename << endl;
        return;
    }

    char buffer[BUFFER];
    cout << "From the file)))" << endl;
    while (fgets(buffer, BUFFER, file) != nullptr) {
        cout << buffer; // Виводимо зчитаний рядок
    }

    fclose(file);
}

```

## Завдання №4:

### VNS Practice Work Task 4 v16

```

vns_practice_work_4_task_yurii_bubelnyki.cpp > main()
1  #include <iostream>
2  #include <ctime> //підключив бібліотеку для srand та rand
3
4  using namespace std;
5
6  const int SIZE = 10; //задаю розмір масив
7
8  void Initialize(int arr[]);
9
10 int main(){
11     srand(time(NULL)); //для генерації псевдовипадкових чисел при повторному запуску програми
12
13     int arr[SIZE]; // створюю масив типу int
14
15     Initialize(arr); // викликаю функцію для обробки масиву
16
17     return 0;
18 }
19 // функція для обрахунку середнього арифметичного послідовності з 10 чисел в діапазоні від 1 до 10
20 void Initialize(int arr[]){
21
22     int sum = 0; // ініціалізація змінної для запису суми послідовності
23
24     for (int i = 0; i < SIZE; i++){
25         arr[i] = rand() % 10 + 1; // обмежую діапазон від [1, 10]
26         sum += arr[i]; // сумую значення послідовності
27         cout << arr[i] << " "; // вивід значень послідовності
28     }
29
30     cout << "\nAverage: " << static_cast<double>(sum) / SIZE << endl; //переведення типу int в double для обчислення середнього арифметичного
31 }
32

```

## Завдання №5:

### Algotester Lab 1v1

```

Lab-1v1.cpp > main()
1  #include <iostream>
2  using namespace std;
3
4  int main() {
5      long long H, M; // long long бо межі 1e12
6      cin >> H >> M;
7
8      if (H < 1 || H > 1e12 || M < 1 || M > 1e12) // перевірка умови
9          return 1;
10     // заповнення значень
11     for (int i = 0; i < 3; i++) {
12         long long hi, mi;
13         cin >> hi >> mi;
14         // перевірка умови чи не виходить за межі необхідна кількість ресурсів на заклинання
15         if (hi < 0 || hi > 1e12 || mi < 0 || mi > 1e12)
16             return 1;
17         // якщо витрачається одразу два показника то програш
18         if (hi > 0 && mi > 0) {
19             cout << "NO" << endl;
20             return 0;
21         }
22         // віднімання ресурсів після заклинання
23         if (hi > 0) {
24             H -= hi;
25         } else if (mi > 0) {
26             M -= mi;
27         }
28         // перевірка на кількість ресурсів
29         if (H <= 0 || M <= 0) {
30             cout << "NO" << endl;
31             return 0;
32         }
33     }
34
35
36     cout << "YES" << endl;
37     return 0;
38 }
39

```

## Завдання №6:

### Algotester Lab 2v1

```

#include <iostream>
#include <vector> // для списку розмір якого визначає користувач
using namespace std;

int main() {
    int N;
    cin >> N;

    if (N > 1e5 || N < 1) {
        return 1;
    }

    if (N <= 2) {
        cout << 0 << endl;
        return 0;
    }

    vector<int> r(N); // список з N елементів

    for (int i = 0; i < N; i++) {
        cin >> r[i];
        if (r[i] > 1e5 || r[i] < 0) {
            return 1;
        }
    }

    // запис початкових значень min та max
    int max = r[0], min = r[0];
    int maxIndex = 0, minIndex = 0;

    for (int i = 1; i < N; i++) {
        if (r[i] > max) {
            max = r[i];
            maxIndex = i;
        }
        if (r[i] < min) {
            min = r[i];
            minIndex = i;
        }
    }

    // нові значення max та min
    int newMax = (maxIndex == 0) ? r[1] : r[0];
    int newMin = (minIndex == 0) ? r[1] : r[0];

    // цикл для знаходження новий значень max та min
    for (int i = 0; i < N; i++) {
        if (i != maxIndex) {
            if (r[i] > newMax) {
                newMax = r[i];
            }
        }
        if (i != minIndex) {
            if (r[i] < newMin) {
                newMin = r[i];
            }
        }
    }

    // різниця між парами елементів
    int rMax = newMax - min;
    int rMin = max - newMin;

    if (rMax < rMin) {
        cout << rMax << endl;
    } else {
        cout << rMin << endl;
    }

    return 0;
}

```

**Завдання №7:**

**Апельсини**

```

algotester_practice_work_1.cpp > ...
1  #include <iostream>
2  using namespace std;
3
4  int main() {
5      long long a, b, c;
6      cin >> a >> b >> c;
7      // перевірка меж
8      if(a < 1 || a > 1e9 || b < 1 || b > 1e9 || c < 1 || c > 1e9) return 1;
9
10     if(a + b > c)
11     {
12         cout << "YES";
13     }
14     else
15     {
16         cout << "NO";
17     }
18
19     return 0;
20 }
21

```

## Завдання №8:

### Algotester Lab 3v3

```

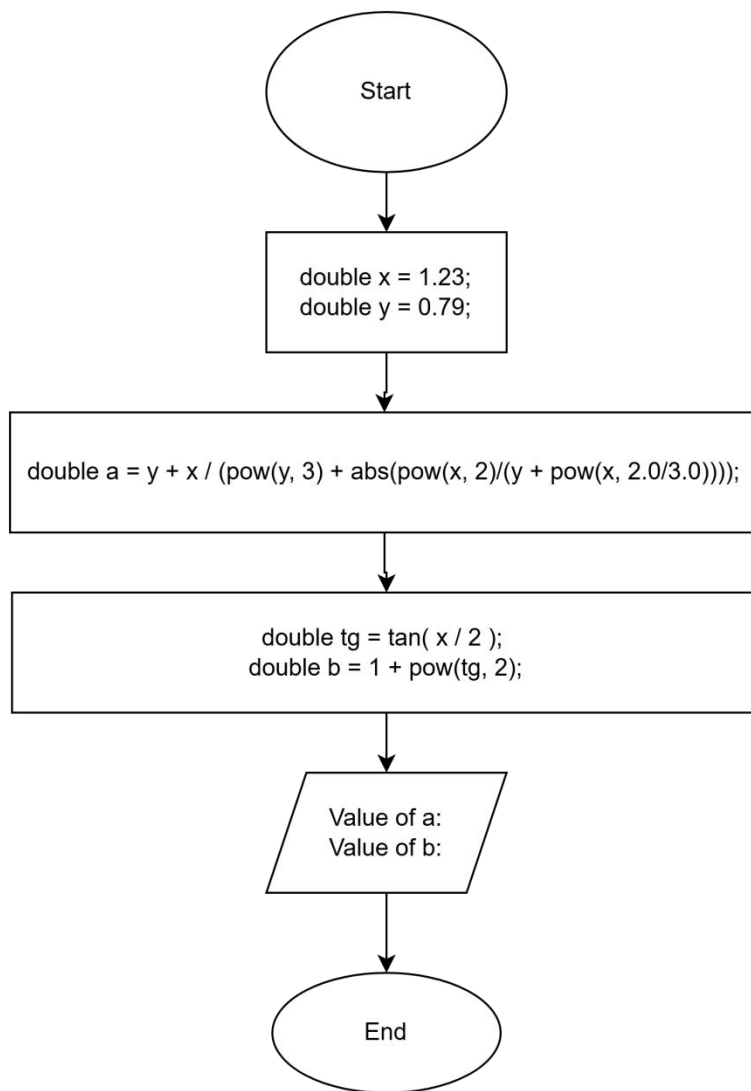
Lab-3v1.cpp > main()
1  #include <iostream>
2  #include <string> // для string, length, to_string
3
4  using namespace std;
5
6  int main() {
7
8      string s, sc; // створення string для запису рядка
9      cin >> s;
10     if(s.length() < 1 || s.length() > 1e5) return 1; // перевірка меж
11
12     int count = 1; // лічильник для запису кількості повторюваних символів
13
14     for(int i = 0; i < s.length(); i++)
15     {
16         if(i + 1 < s.length() && s[i] == s[i + 1])
17         {
18             count++;
19         }else{
20             sc += s[i]; // додавання символів
21             if(count == 1) continue;
22             sc += to_string(count); // запис числа як символ
23             count = 1;
24         }
25     }
26
27     cout << sc << endl; // вивід рядка
28
29     return 0;
30 }

```

## Дизайн та планована оцінка часу виконання завдань:

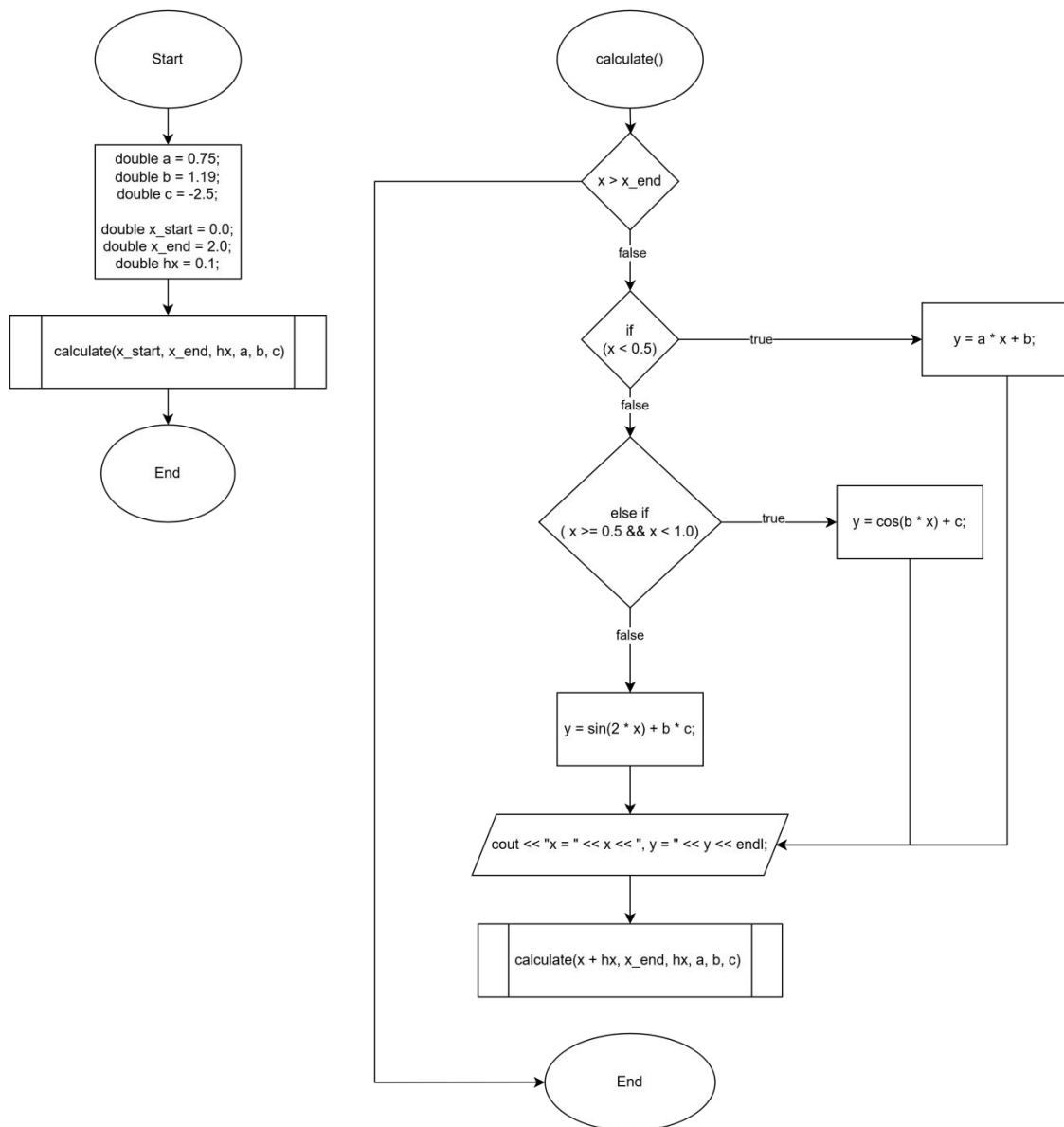
### Завдання №1 – VNS Practice Work Task 1 v25

Планований час виконання 10хвилин.



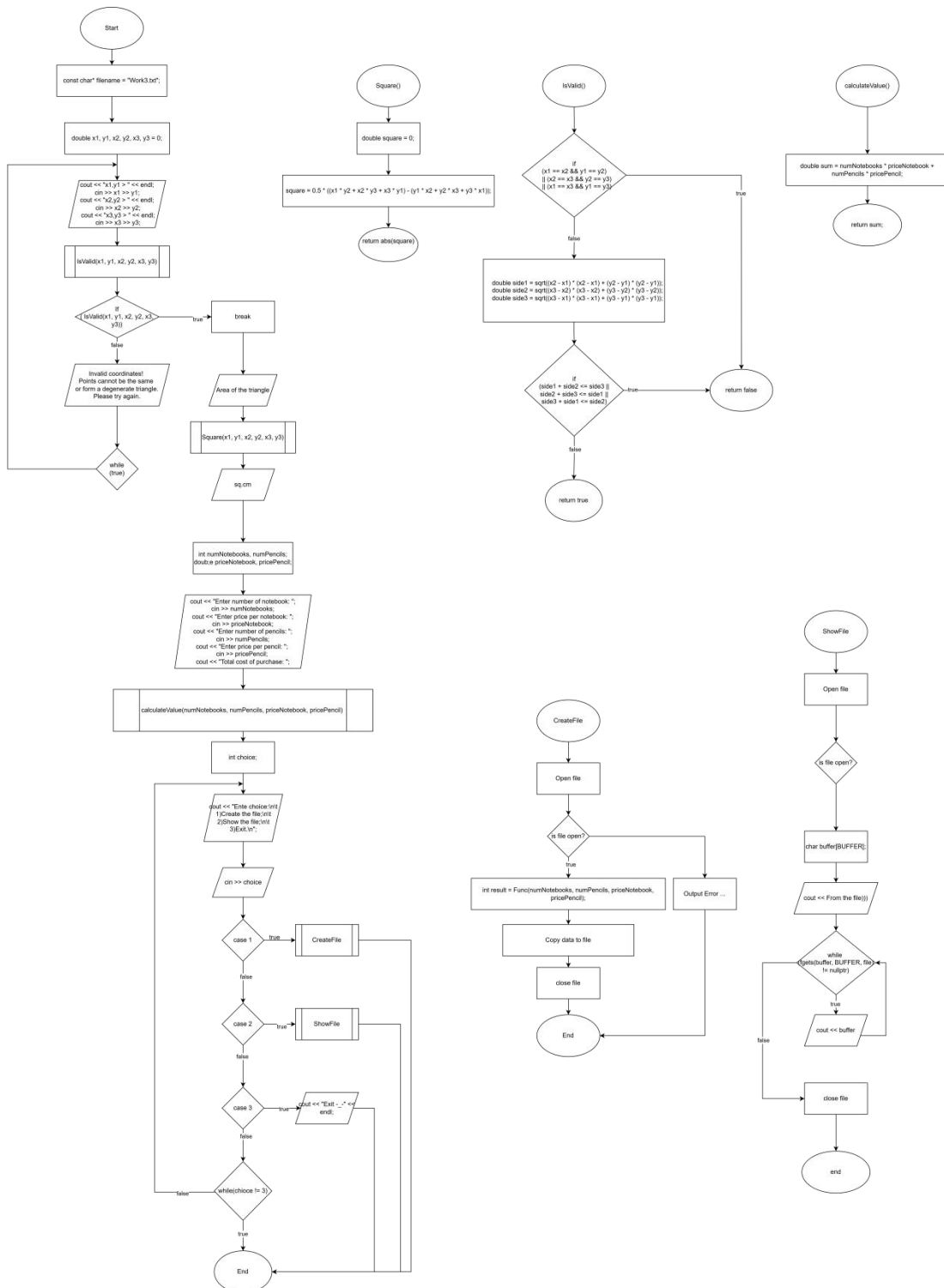
## Завдання №2 – VNS Practice Work Task 2 v1

Планований час виконання 15хвилин.



### Завдання №3 – VNS Practice Work Task 3 v19

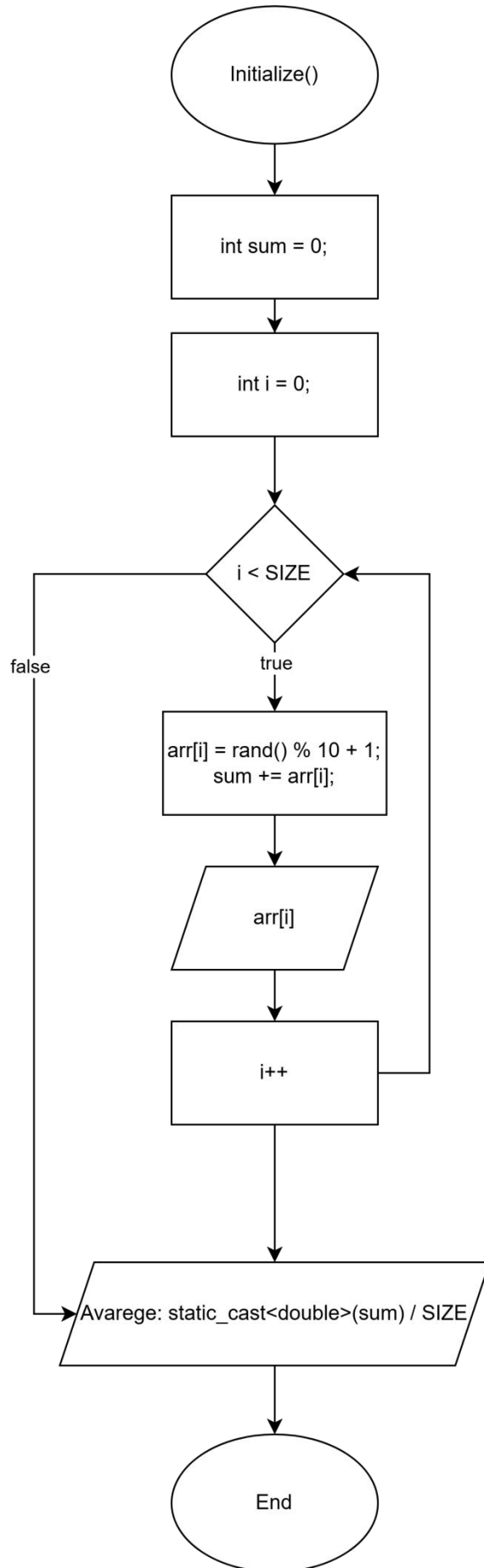
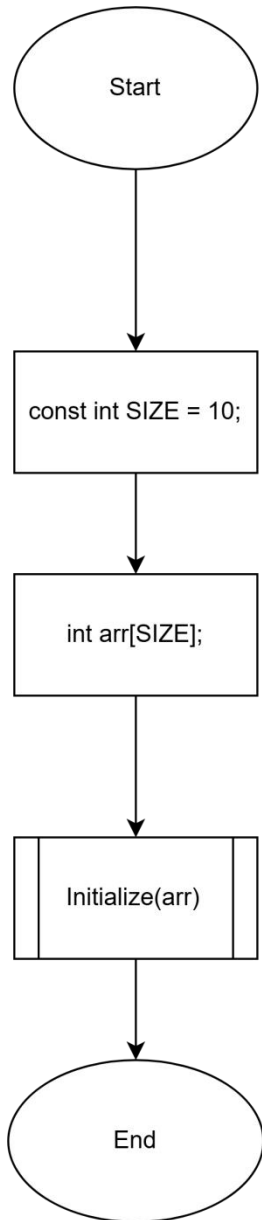
Планований час виконання 1 година.



## Завдання №4 – VNS Practice Work Task 4 v16

Планований час виконання 15хвилин.

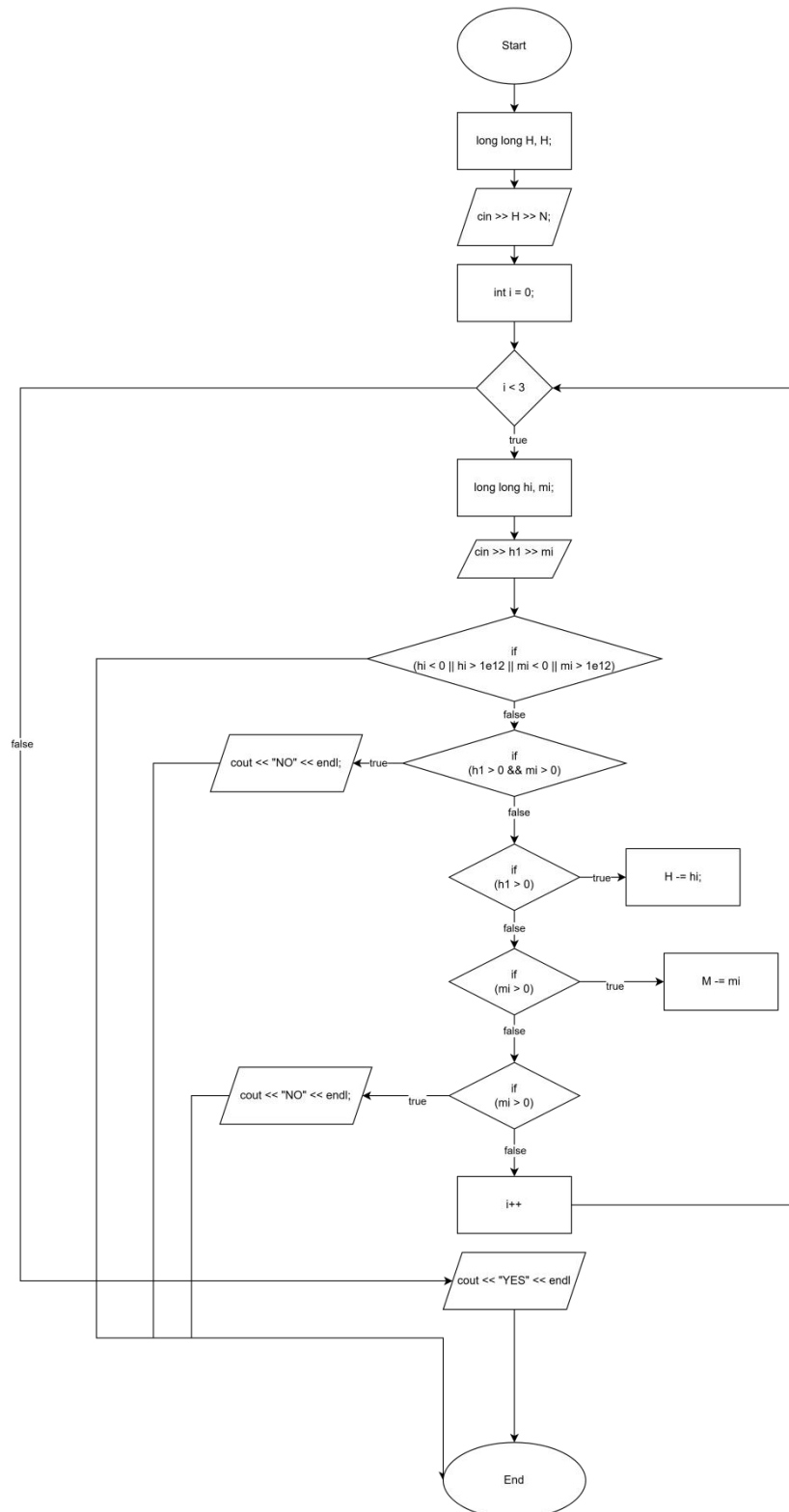




## Завдання №5:

### Algotester Lab 1v1

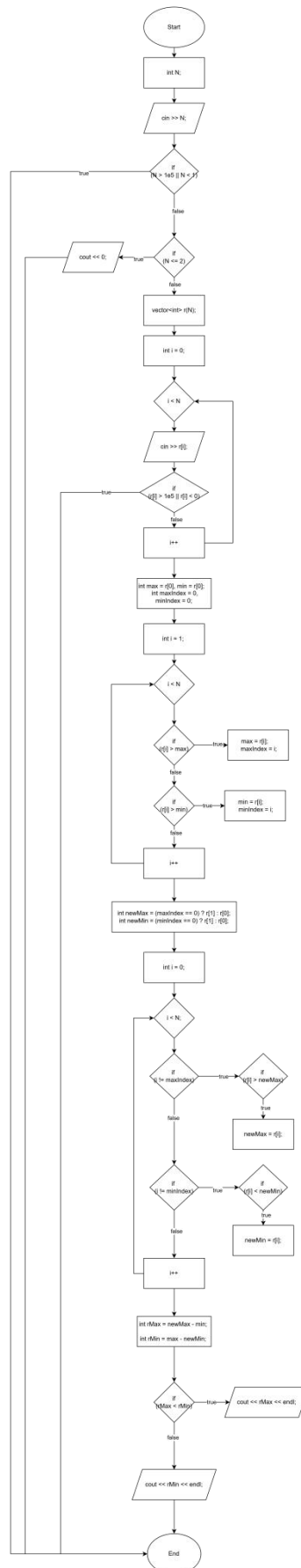
Планований час виконання 15хвилин.



## Завдання №6:

### Algotester Lab 2v1

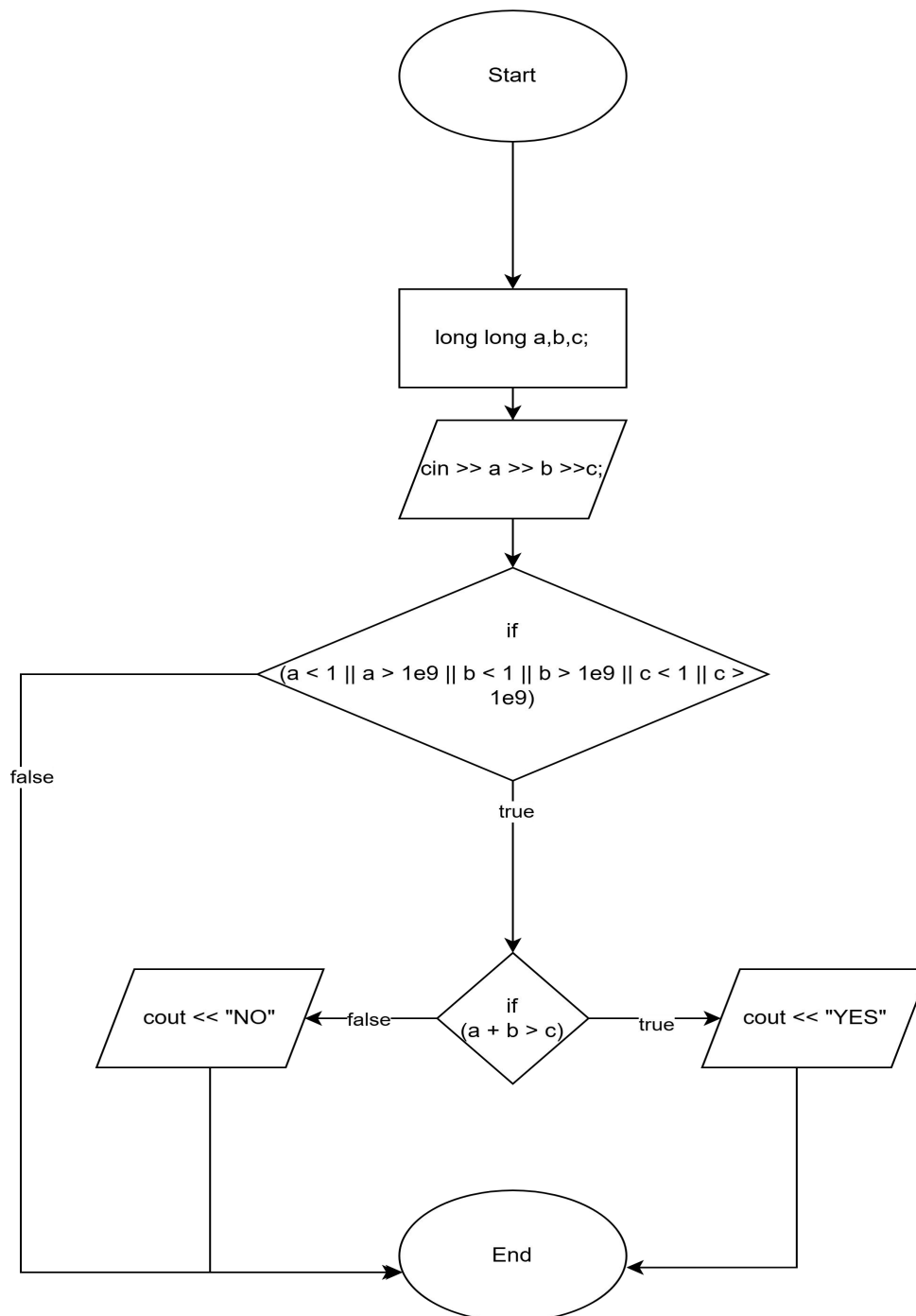
Планований час виконання 40хвилин.



## Завдання №7

### Апельсини

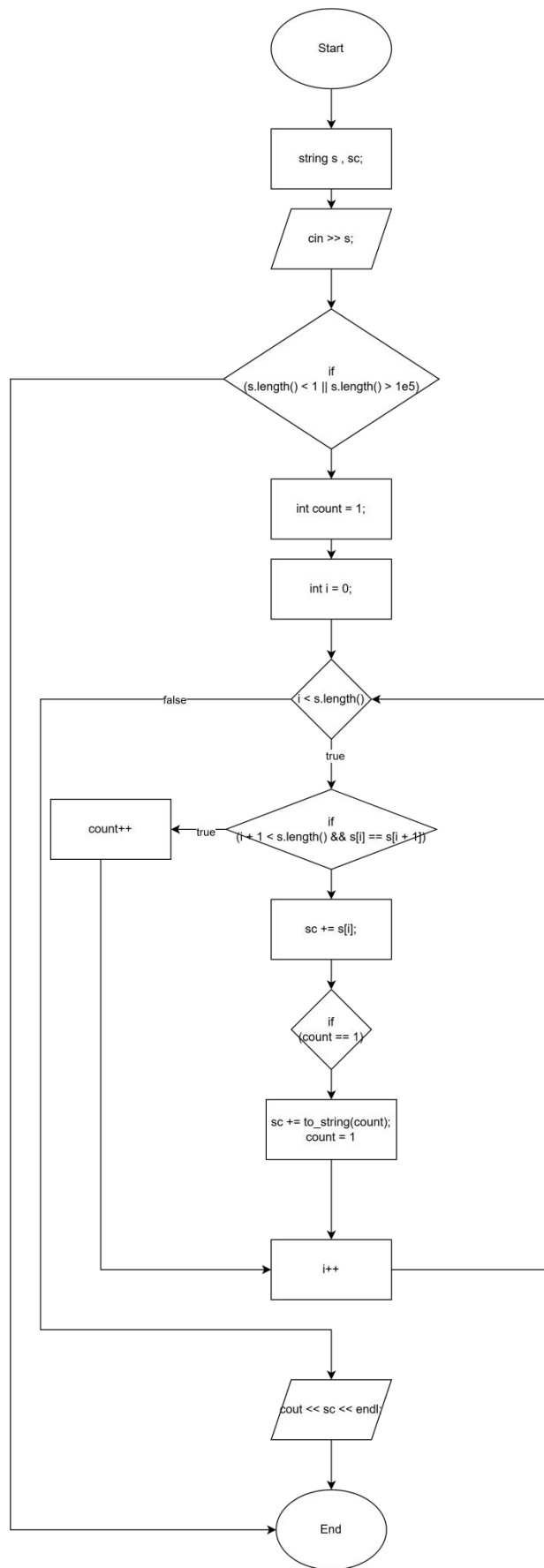
Планований час виконання 5 хвилин.



## Завдання №8

### Algotester Lab 3v3

Планований час виконання 1 години.



Код програми з посиланням на зовнішні ресурси

[Завдання №1 – VNS Practice Work Task 1 v25](#)

[Завдання №2 – VNS Practice Work Task 2 v1](#)

[Завдання №3 – VNS Practice Work Task 3 v19](#)

[Завдання №4 – VNS Practice Work Task 4 v16](#)

[Завдання №5 Algotester Lab 1v1](#)

[Завдання №6 Algotester Lab 2v1](#)

[Завдання №7 Апельсини](#)

[Завдання №8 Algotester Lab 3v3](#)

**Результати виконаних завдань, тестування та фактично  
затрачений час**

**Завдання №1 – VNS Practice Work Task 1 v25**

**~10 хвилин**

```
PS C:\Users\Admin\Desktop\Saga-1> cd "c:\Users\Admin\Desktop\Saga-1\"
Value of a = 1.75569          Value of b = 1.49898

PS C:\Users\Admin\Desktop\Saga-1>
```

**Завдання №2 – VNS Practice Work Task 2 v1**

**~15 хвилин**

```
PS C:\Users\Admin\Desktop\Saga-1> cd "c:\Users\Admin\Desktop\Saga-1\"
x = 0, y = 1.19
x = 0.1, y = 1.265
x = 0.2, y = 1.34
x = 0.3, y = 1.415
x = 0.4, y = 1.49
x = 0.5, y = -1.67185
x = 0.6, y = -1.74425
x = 0.7, y = -1.82734
x = 0.8, y = -1.91994
x = 0.9, y = -2.02075
x = 1, y = -2.12834
x = 1.1, y = -2.1665
x = 1.2, y = -2.29954
x = 1.3, y = -2.4595
x = 1.4, y = -2.64001
x = 1.5, y = -2.83388
x = 1.6, y = -3.03337
x = 1.7, y = -3.23054
x = 1.8, y = -3.41752
x = 1.9, y = -3.58686
PS C:\Users\Admin\Desktop\Saga-1>
```

**Завдання №3 – VNS Practice Work Task 3 v19**

~1 година

```
PS C:\Users\Admin\Desktop\Saga-1> cd "c:\Users\Admin\Desktop\Saga-1\" ; if ($?) {
x1,y1 >
2 2
x2,y2 >
3 7
x3,y3 >
7 9
Area of the triangle: 9.00 sq.cm.

Enter number of notebooks: 10
Enter price per notebook: 10
Enter number of pencils: 10
Enter price per pencil: 10
Total cost of purchase: 200.00 UAH.
Enter choice:
    1)Create the file;
    2)Show the file;
    3)Exit.

1
Result has been written to the file: Work3.txt
Enter choice:
    1)Create the file;
    2)Show the file;
    3)Exit.

2
From the file)))
The total value is: 200
Enter choice:
    1)Create the file;
    2)Show the file;
    3)Exit.

3
Exit -_-
PS C:\Users\Admin\Desktop\Saga-1> 
```

#### Завдання №4 – VNS Practice Work Task 4 v16

~25 хвилин

```
PS C:\Users\Admin\Desktop\Saga-1> cd "c
7 1 9 6 9 2 3 3 3 3
Average: 4.6
PS C:\Users\Admin\Desktop\Saga-1> 
```

#### Завдання №5 Algotester Lab 1v1

~10 хвилин

9 days ago	<a href="#">Lab 1v1 - Lab 1v1</a>	C++ 23	Accepted	0.003	1.207	1884952
------------	-----------------------------------	--------	----------	-------	-------	---------

#### Завдання №6 Algotester Lab 2v1

~15 хвилин

9 days ago	<a href="#">Lab 2v1 - Lab 2v1</a>	C++ 23	Accepted	0.003	1.211	1884962
------------	-----------------------------------	--------	----------	-------	-------	---------

#### Завдання №7 Апельсини



~4 хвилин

28 minutes ago	<a href="#">0481 - Апельсини</a>	C++ 23	Accepted	0.002	2.020	<a href="#">1907483</a>
----------------------	----------------------------------	--------	----------	-------	-------	-------------------------

## Завдання №8 Algotester Lab 3v3

~30 хвилин

8 days ago	<a href="#">Lab 3v3 - Lab 3v3</a>	C++ 23	Accepted	0.003	1.570	<a href="#">1885766</a>
---------------	-----------------------------------	--------	----------	-------	-------	-------------------------

### Висновки:

Я покращив свої навички, практикуючись на написанні програм необхідних для розрахункової, а також структурував весь пройдений матеріал.

[Посилання на pull request](#)