

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



Звіт
про виконання розрахункових та практичних робіт
блоку № 1
з дисципліни: «Основи програмування»

Виконав:

Студент групи ШІ-11

Лопатін Володимир Дмитрович

Львів 2024

Тема:

Реалізація рішень до практичних завдань з використанням основних принципів, функцій та алгоритмів структурного програмування.

Мета роботи:

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

Виконання роботи:

- 1) Опрацювання завдання та вимог до програм та серидовища:

Завдання №1 з ВНС

Варіант 7. $S = \frac{x^3 \lg^2(x+b)^2 + a}{\sqrt{x+b}} c$; $c = \frac{bx^2 - c}{e^{ax} - 1}$, де $a=16,5$; $b=3,4$; $x=0,61$.

Завдання №2 з ВНС

Варіант 19. Підрахувати, скільки разів функція $y = \cos x^2 \cdot e^{-x}$ приймає негативне значення, якщо $x \in [0,3;5]$; $h_x = 0,1$.

Завдання №3 з ВНС

Варіант 12. Обчислити швидкість, з якою бігун пробіг дистанцію. Нижче приведений вид екрану під час виконання програми, що рекомендується (дані, які вводяться користувачем,

14

виділені напівжирним шрифтом).

Обчислення швидкості бігу.

Введіть довжину дистанції (метрів) > **1000**

Ведіть час (хвилин.секунд) > **3.25**

Дистанція: 1000 м

Часи 3 мін 25 сік = 205 сік

Швидкості бігу = 17.56 км/година

Обчислення опору електричної цінуї по відомих значеннях напруги сили струму.

Завдання №4 з ВНС

Варіант 14. Скласти програму, яка вводить з клавіатури послідовність з п'яти дробових чисел і після введення кожного

20

числа виводить середнє арифметичне введеній частині послідовності.

Завдання №5 з Algotester

«Непарний масив»

Зеник і Марічка мають масив з n цілих чисел a_i . Вони хочуть перевпорядкувати елементи масиву так, щоб сума будь-яких двох послідовних елементів була непарною.

Будь ласка, допоможіть їм зробити це або скажіть, що це неможливо.

Вхідні дані

Перший рядок містить ціле число n — розмір масиву.

Другий рядок містить n цілих чисел a_i — елементи масиву.

Вихідні дані

У єдиному рядку виведіть n цілих чисел — перевпорядкований масив такий, що сума будь-яких двох послідовних елементів є непарною, або **-1**, якщо так зробити неможливо.

Завдання №6 з Algotester

«Дракон і дим»

Потрібно було знайти відстань, на яку видно дим у залежності від заданих параметрів координат та кількостей пожеж, які побачили та загалом.

Завдання №7 з Algotester

«Піраміда та куби»

Виконав завдання, де потрібно було з урахуванням умов та обмежень сказати, чи зможе персонаж користувача поставити пірадміду з п'яти кубів.

Завдання №8 з Algotester

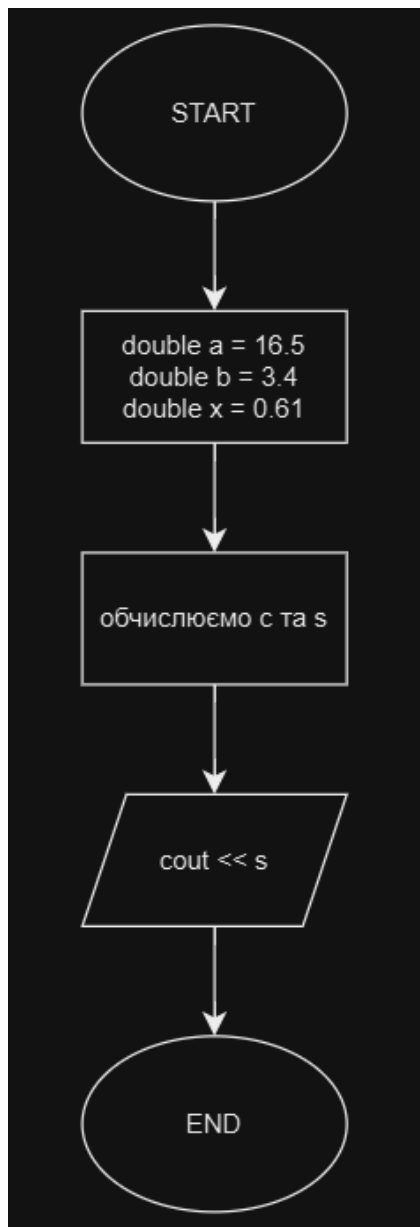
«Спекотні дні пінгвінів»

Тут потрібно скласти умову, яка б перевірила, чи пінгвін зможе випити коктейль, та врахувати вимоги щодо чисел, які можуть бути введені та вихідних даних.

2) Дизайн та планова оцінка часу виконання завдань:

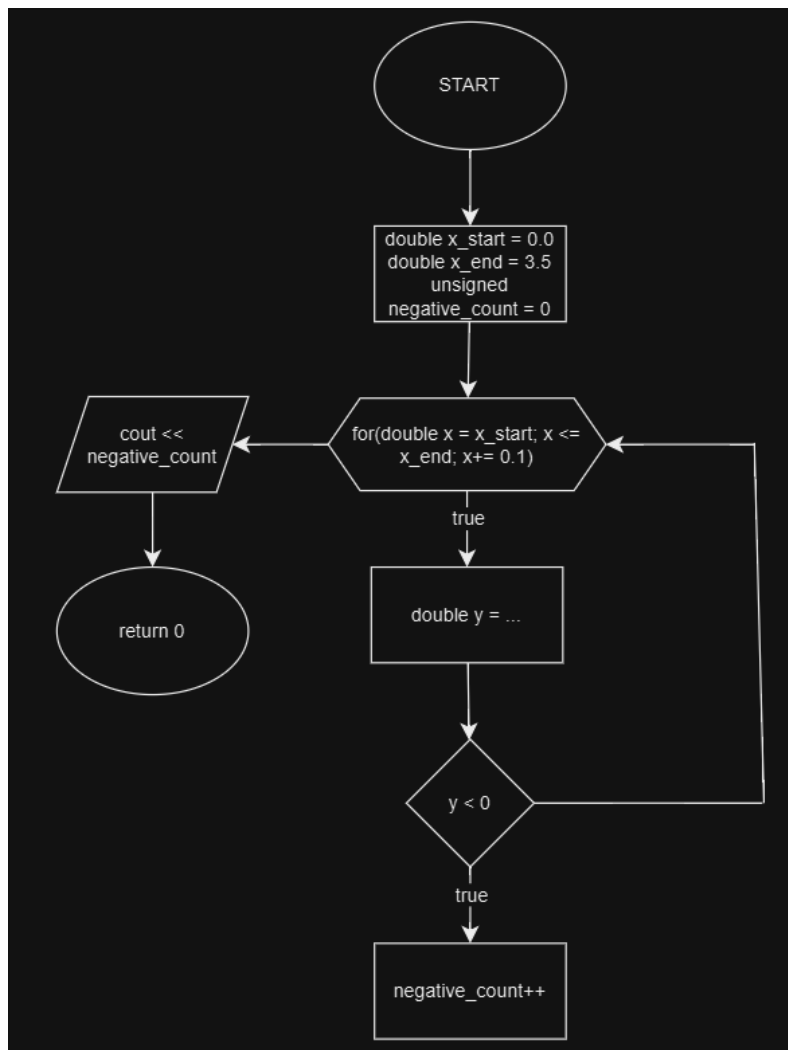
Завдання №1 з ВНС

Планував виконати за 20 хвилин.



Завдання №2 з ВНС

Планував витратити 15 хвилин.



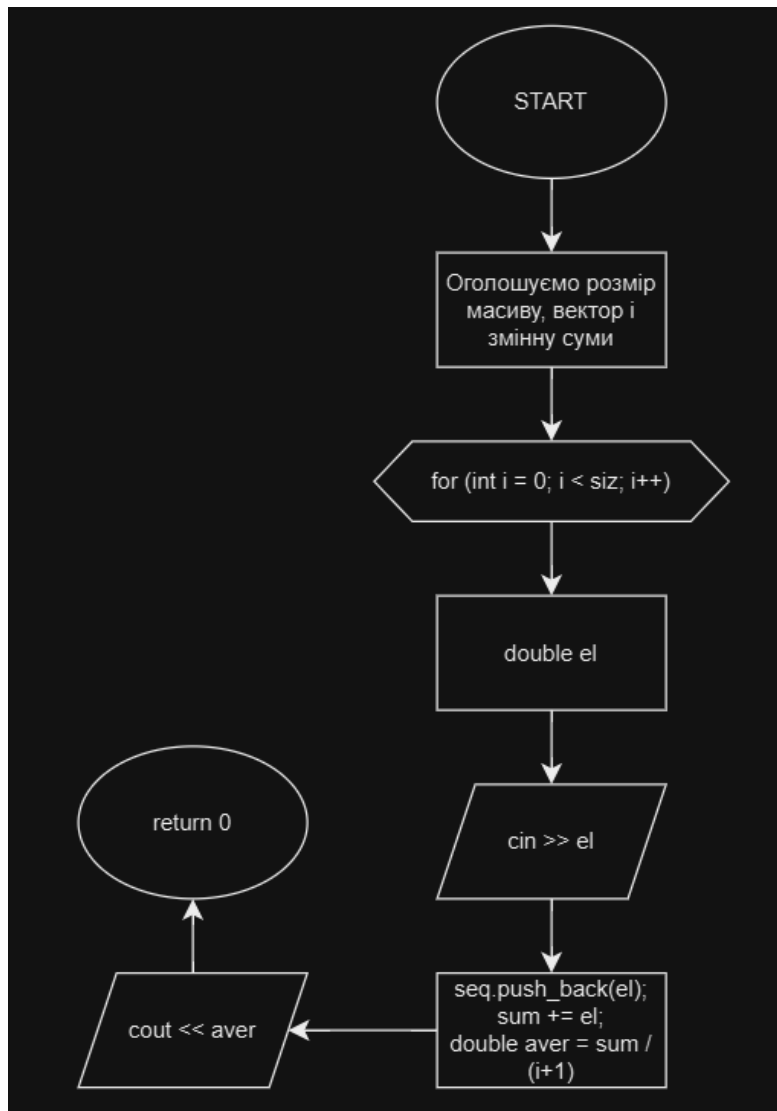
Завдання №3 з ВНС

Планував витратити 20 хвилин.

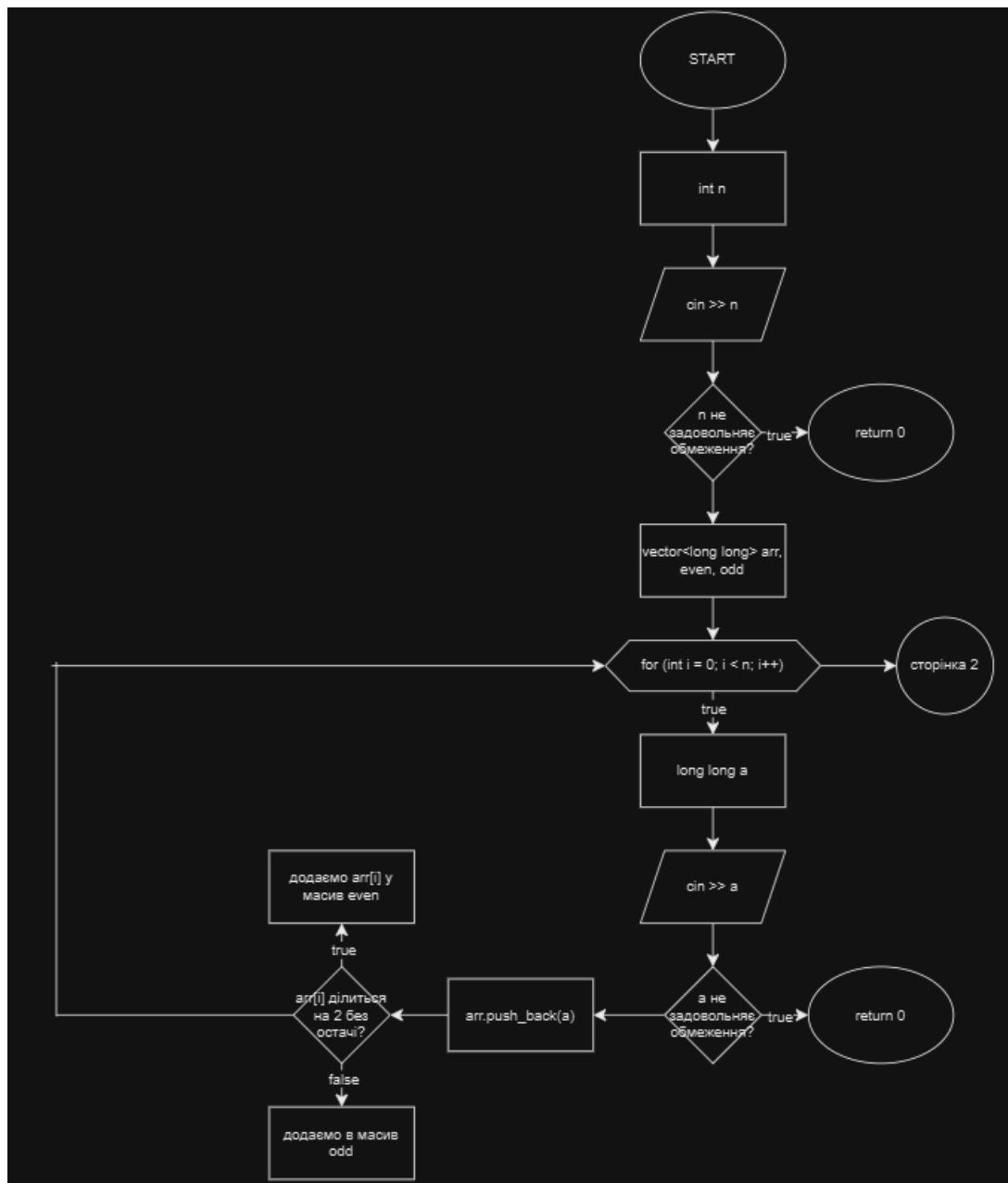


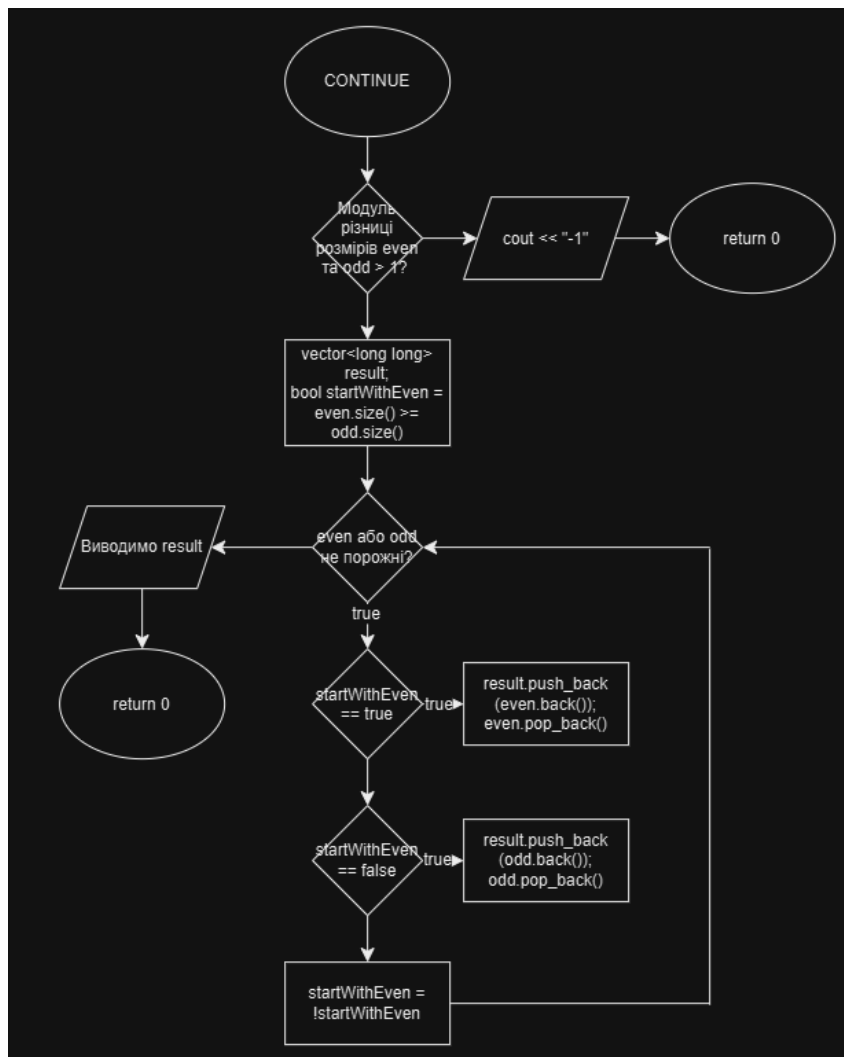
Завдання №4 з ВНС

Планував зробити за 20 хвилин.



Завдання №5 з Algotester

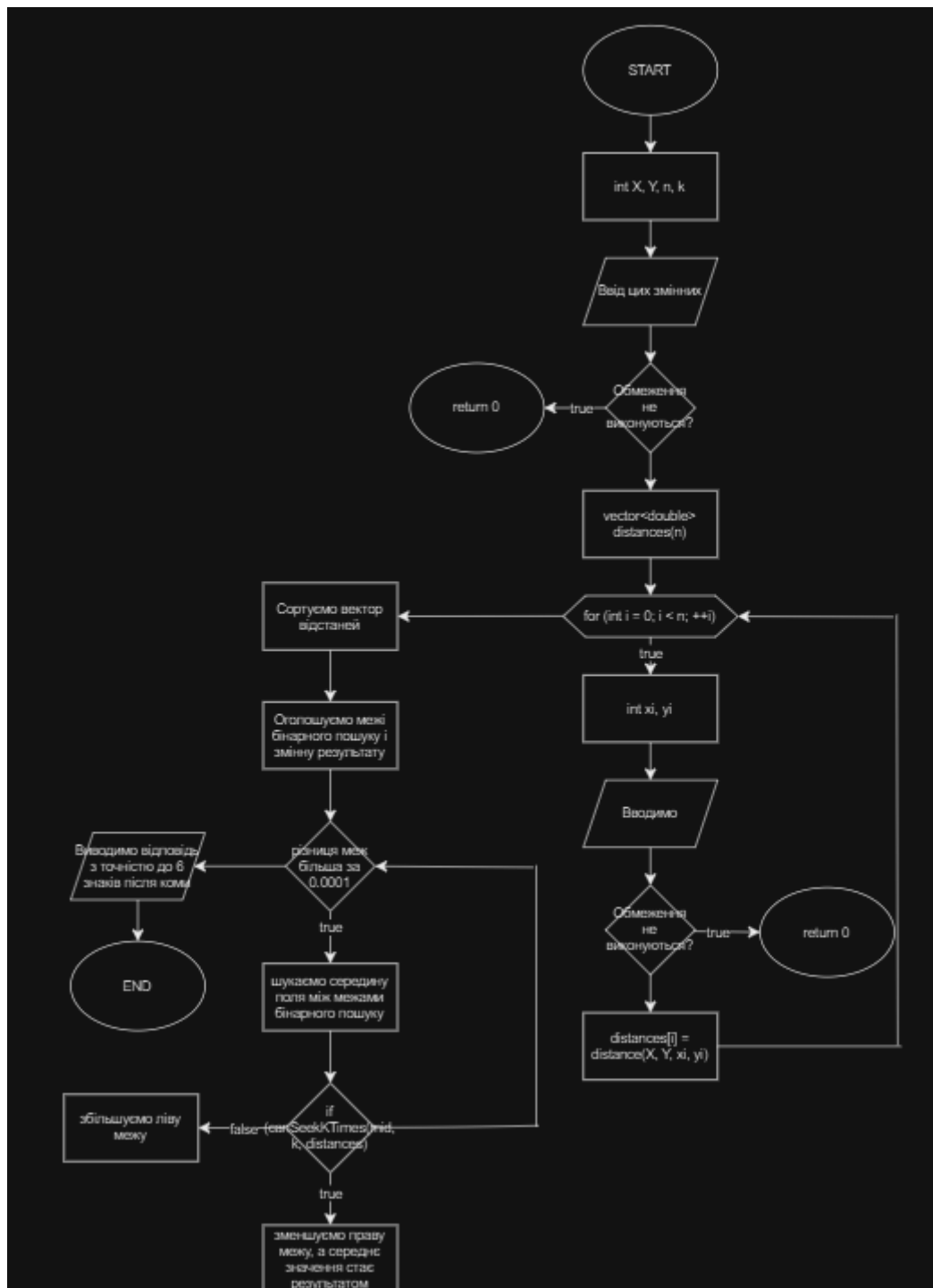




Орієнтувався виконати за 2 години.

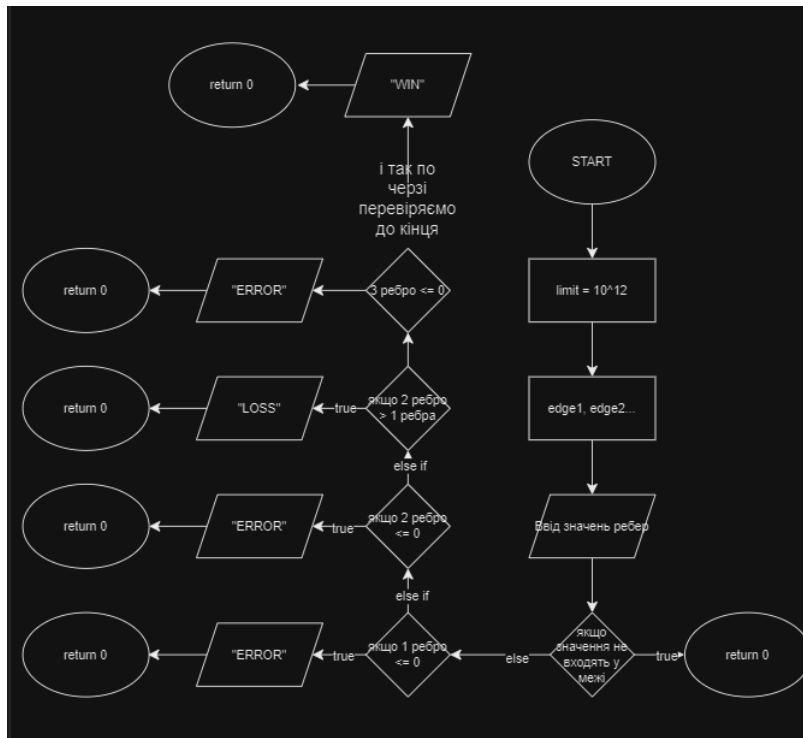
Завдання №6 з Algotester

На це завдання я планував витратити 2 години разом із діаграмою.



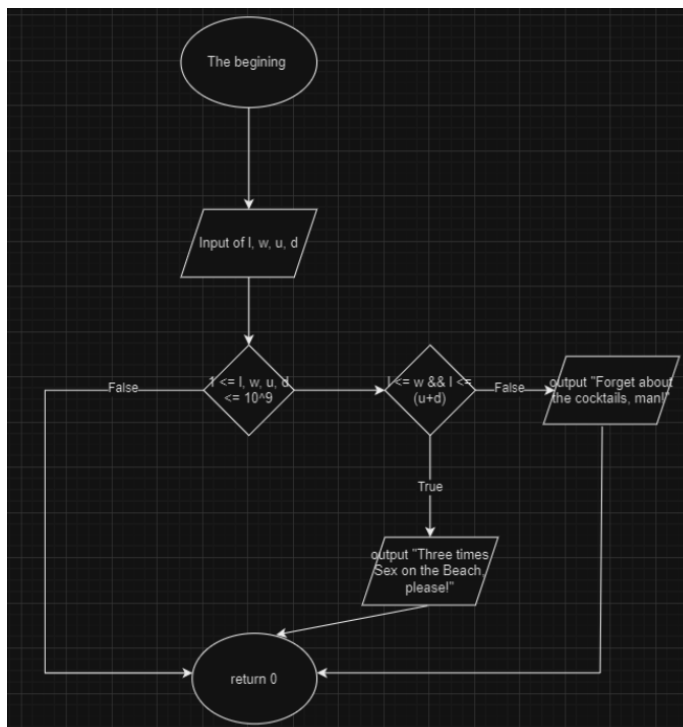
Завдання №7 з Algotester

На це завдання планував витратити не більше 2,5 годин.



Завдання №8 з Algotester

На це завдання я планував затратити 2-3 години, але зайняло трохи більше часу.



3) Код програм з посиланням на зовнішні ресурси:

Завдання №1

```
#include <iostream>
#include <cmath>
using namespace std;
int main() {
    double a = 16.5;
    double b = 3.4;
    double x = 0.61;
    double c = (b*x*x)/(exp(a*x) - 1.0);
    double s = (x*x*x*pow(tan((x+b)*(x+b)), 2) + a)*c / sqrt(x+b);
    cout << "S = " << s;
    return 0;
}
```

Завдання №2

```
#include <iostream>
#include <cmath>
using namespace std;
int main() {
    double x_start = 0.0;
    double x_end = 3.5;
    unsigned negative_count = 0;

    for (double x = x_start; x <= x_end; x += 0.1) {
        double y = cos(x * x) * exp(-x);
        if (y < 0) {
            negative_count++;
        }
    }

    cout << "Кількість від'ємних значень функції: " << negative_count << endl;

    return 0;
}
```

Завдання №3

```
#include <iostream>
using namespace std;

int main() {
    double dist;
    double time;
    double coef = 0.06;
    cout << "Enter the distance: ";
    cin >> dist;
    cout << "Enter the time: ";
    cin >> time;
    cout << "The velocity is " << dist*coef / time << "km/hr" << endl;

    double voltage, amperage;
    cin >> voltage >> amperage;
    cout << "The resistance is " << voltage/amperage << " Ohm";

    return 0;
}
```

Завдання №4

```
#include <iostream>
#include <vector>
using namespace std;

int main() {
    int siz = 5;
    vector<double> seq;
    double sum = 0.0;
    for (int i = 0; i < siz; i++) {
        double el;
        cin >> el;
        seq.push_back(el);
        sum += el;
        double aver = sum / (i + 1);
        cout << "Середнє арифметичне: " << aver << endl;
    }

    return 0;
}
```

Завдання №5

```
#include <iostream>
#include <vector>
#include <cmath>
using namespace std;

int main() {
    int n;
    cin >> n;
    if (n < 1 || n > 100000) return 0;
    vector<long long> arr;
    vector<long long> even, odd;
    for (int i = 0; i < n; ++i) {
        long long a;
        cin >> a;
        if (a < 1 || a > 1000000000) return 0;
        arr.push_back(a);
        if (arr[i] % 2 == 0) {
            even.push_back(arr[i]);
        } else {
            odd.push_back(arr[i]);
        }
    }
    if (abs((int)even.size() - (int)odd.size()) > 1) {
        cout << "-1";
        return 0;
    }
    vector<long long> result;
    bool startWithEven = even.size() >= odd.size();
    while(!odd.empty() || !even.empty()) {
        if (!even.empty() && startWithEven) {
            result.push_back(even.back());
            even.pop_back();
        } else if (!odd.empty() && !startWithEven) {
            result.push_back(odd.back());
            odd.pop_back();
        }
        startWithEven = !startWithEven;
    }
    for (long long num : result) {
        cout << num << " ";
    }
    return 0;
}
```


Завдання №6

```
#include <iostream>
#include <vector>
#include <cmath>
#include <algorithm>

using namespace std;

double distance(double x1, double y1, double x2, double y2) {
    return sqrt((x1 - x2) * (x1 - x2) + (y1 - y2) * (y1 - y2));
}

bool canSeeKTimes(double r, int k, const vector<double>& distances) {
    int count = 0;
    for (double d : distances) {
        if (d <= r) {
            count++;
        }
        if (count >= k) {
            return true;
        }
    }
    return false;
}

int main() {
    int X, Y;
    int n, k;
    cin >> X >> Y;
    cin >> n >> k;

    if (X < 0 || X > 1e9 || Y < 0 || Y > 1e9) {
        return 0;
    }

    if (n < 1 || n > 100 || k < 1 || k > n) {
        return 0;
    }

    vector<double> distances(n);
    for (int i = 0; i < n; i++) {
        int xi, yi;
        cin >> xi >> yi;
        if (xi < 0 || xi > 1e9 || yi < 0 || yi > 1e9) {
            return 0;
        }
        distances[i] = distance(X, Y, xi, yi);
    }

    sort(distances.begin(), distances.end());

    double left = 0.0, right = distances.back();
    double answer = right;

    while (right - left > 1e-4) {
        double mid = (left + right) / 2.0;
        if (canSeeKTimes(mid, k, distances)) {
            answer = mid;
            right = mid;
        } else {
            left = mid;
        }
    }

    cout.precision(6);
    cout << fixed << answer << endl;

    return 0;
}
```

Завдання №7

```
#include <iostream>
using namespace std;

int main() {
    const long long LIMIT = 1000000000000;
    long long edge1, edge2, edge3, edge4, edge5;

    cin >> edge1 >> edge2 >> edge3 >> edge4 >> edge5;

    if (edge1 <= -LIMIT || edge1 >= LIMIT ||
        edge2 <= -LIMIT || edge2 >= LIMIT ||
        edge3 <= -LIMIT || edge3 >= LIMIT ||
        edge4 <= -LIMIT || edge4 >= LIMIT ||
        edge5 <= -LIMIT || edge5 >= LIMIT) {
        return 0;
    } else {
        if (edge1 <= 0) {
            cout << "ERROR" << endl;
            return 0;
        } else if (edge2 <= 0) {
            cout << "ERROR" << endl;
            return 0;
        } else if (edge2 > edge1) {
            cout << "LOSS" << endl;
            return 0;
        }

        if (edge3 <= 0) {
            cout << "ERROR" << endl;
            return 0;
        } else if (edge3 > edge2) {
            cout << "LOSS" << endl;
            return 0;
        }

        if (edge4 <= 0) {
            cout << "ERROR" << endl;
            return 0;
        } else if (edge4 > edge3) {
            cout << "LOSS" << endl;
            return 0;
        }

        if (edge5 <= 0) {
            cout << "ERROR" << endl;
            return 0;
        } else if (edge5 > edge4) {
            cout << "LOSS" << endl;
            return 0;
        }
    }

    cout << "WIN" << endl;
    return 0;
}
```

Завдання №8

```
D:\> epic1 > G:\self_practice_work_algotester_task_1_volodymyr_lopatin.cpp > ...
1  #include <iostream>
2  using namespace std;
3
4  int main() {
5      unsigned l, w, u, d;
6      cin >> l >> w >> u >> d;
7      if ( l >= 1 && l <= 1000000000 && w >= 1 && w <= 1000000000 && u >= 1 && u <= 1000000000 && d >= 1 && d <= 1000000000){
8          if (l <= w && l <= (u + d)) {
9              cout << "Three times Sex on the Beach, please!";
10         } else {
11             cout << "Forget about the cocktails, man!";
12         }
13     }
14
15     return 0;
16 }
```

4) Результати виконання завдань, тестування та фактично
затрачений час

Завдання №1

```
S = 0.000444431
PS D:\>
```

Витратив 15 хвилин.

Завдання №2

```
Кількість від'ємних значень функції: 14  
PS D:\>
```

Витратив на завдання близько 15 хвилин.

Завдання №3

На це завдання пішло 25 хвилин.

```
Enter the distance: 1400  
Enter the time: 5.5  
The velocity is 15.2727km/hr  
5 3  
The resistance is 1.66667 Om  
PS D:\>
```

Завдання №4

```
1  
Середнє арифметичне: 1  
2  
Середнє арифметичне: 1.5  
3  
Середнє арифметичне: 2  
6  
Середнє арифметичне: 3  
13  
Середнє арифметичне: 5  
PS D:\>
```

Витратив на завдання приблизно 30 хвилин.

Завдання №5

```
5  
3 6 8 3 9  
9 8 3 6 3  
PS D:\>
```

Опрацював завдання годину.

Завдання №6

```
5 6  
4 2  
3 4  
8 11  
6 5  
17 25  
2.828432
```

Витратив на завдання близько 1 години.

Завдання №7

```
9 8 7 6 5  
WIN  
PS D:\> c
```

Ось варіант, коли виводить «WIN».

```
4 3-2 6 9  
ERROR  
PS D:\> c
```

Ось, коли «ERROR».

```
3  
2  
1  
8  
0  
LOSS  
PS D:\> █
```

І варіант вводу, коли на виході отримуємо «LOSS».

На це завдання пішло 6 годин.

Завдання №8

```
PS D:\> & 'c:\Users\Admin\.vscode\extensions\ms-vscode.cpptools-1.21.6-win32-x64\
=Microsoft-MIEngine-Out-3ad1revh.eev' '--stderr=Microsoft-MIEngine-Error-ddfnzzev.
reter=mi'
7
9
5
6
Three times Sex on the Beach, please!
PS D:\> █
```

Це умова повертає 1.

```
PS D:\> & 'c:\Users\Admin\.vscode\extensions\ms-vscode.cpptools-1.21.6-win32-x64\
=Microsoft-MIEngine-Out-5zjl4f2k.5ze' '--stderr=Microsoft-MIEngine-Error-ddfnzzev.
reter=mi'
10 9 4 5
Forget about the cocktails, man!
PS D:\> █
```

А тут 0.

Висновки:

У ході розрахункової роботи було закріплено знання та навички використання базових алгоритмів, функцій і принципів структурного програмування. Реалізовано програми

для роботи з масивами, сортування, пошуку та умовними операторами. Акцент зроблено на структурному підході, що забезпечує зрозумілість, ефективність і модульність коду. Завдання сприяли глибшому розумінню вибору алгоритмів, оптимізації ресурсів і тестування програм. Робота досягла поставлених цілей і створила основу для подальшого вивчення складніших концепцій програмування.