

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

Виконала:

Студентка групи ІІІ-13
Осінна Єлизавета Сергіївна

Львів 2024

Мета роботи:

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

Виконання роботи:

1. Опрацювання завдання та вимог до програм та середовища:

VNS Practice Work – Task 1

- Варіант 5
- Деталі завдання:

Розробити лінійний алгоритм для розв'язання задачі.

$$\text{Варіант 5. } y = e^{-bt} \sin(at + b) - \sqrt{|bt + a|}; \quad s = b \sin(at^2 \cos 2t) - 1,$$

де $a = -0,5$; $b = 1,7$; $t = 0,44$.

VNS Practice Work – Task 2

- Варіант 15
- Деталі завдання

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

$$\text{Варіант 15. } z = \frac{\sqrt{x-1,5+x^a}}{(x-2)^{1/3}}; \quad x \in [1,4]; \quad h_x = 0,5; \quad a > -0,5;$$

$h_a = 0,2$, де x і a змінюються одночасно.

VNS Practice Work – Task 3

- Варіант 20
- Деталі завдання

Написати програму згідно свого варіанту.

Варіант 5. Обчислення вартості покупки складається з декількох зошитів і такої ж кількості обкладинок до них. Нижче приведений вид екрану під час виконання програми, що рекомендується (дані, які вводяться користувачем, виділені напівжирним шрифтом).

Обчислення вартості покупки.

Введіть початкові дані:

Ціна зошита (грн.) **> 2.75**

Ціна обкладинок (грн.) **> 0.5**

Кількість комплектів (грн.) **> 7**

Вартість покупки: 15.45 грн.

Обчислення об'єму і площі поверхні кулі, радіусом r , за формулами

$$V = \frac{3}{4} \cdot \pi \cdot r^3; \quad S = 4 \cdot \pi \cdot r^2.$$

VNS Practice Work – Task 4

- Варіант 21
- Деталі завдання

Варіант 21. Скласти програму, яка виводить на екран таблицю множення, наприклад, на 7.

Algotester Task – Task 5

- Варіант завдання
- Деталі завдання

День програміста

Нарешті ми його дочекалися, 256-го дня в році (дня програміста)... Кожен зі студентів факультету прикладної математики та інформатики святкували його по-різному: дехто вдома, дехто в «Picasso», дехто в гуртожитку... Не дивно, що Зеник із Марічкою залишилися святкувати його в гуртожитку. Вони організували «mega party» :-). Свято вдалося...

Наступного дня, гуляючи з Марічкою, Зенику стало цікаво, скільки ж було випито різної випивки?!

Марічка змогла згадати nn назв випивок, а Зеник — mm .

Вам потрібно написати програму, яка порахує скільки ж було різної випивки на святі, яку змогли згадати Зеник з Марічкою.

Вхідні дані

У першому рядку дано цілі числа nn і mm .

У наступних nn рядках дано назви випивок, які змогла згадати Марічка.

В наступних mm рядках дано назви випивок, які зміг згадати Зеник.

Вихідні дані

Єдине ціле число — кількість різної випивки на святі.

Обмеження

$0 \leq n, m \leq 777$,

$1 \leq |name_i| \leq 7$ (лише маленькі латинські літери),

$|a|$ — довжина a .

Algotester Task – Task 6

- Деталі завдання

Марічка і печиво

Зібралися Зеник і Марічка разом з пластунами в похід. Похід — серйозна справа. Потрібно заpastись продуктами харчування та розподілити їх споживання по днях так, щоб всім вистачило. Цього разу Зеник слідкує за тим, щоб печива вистачило аж до останнього дня походу. Зеник чітко знає, скільки пачок печива повинно залишитись кожного дня, і щовечора перераховує їх. Якщо Зеник побачить, що залишилось менше пачок, ніж повинно залишитись за його розрахунками, він неодмінно знайде того, хто з'їв забагато печива, і покарає його.

Марічка дуже любить печиво. Сьогодні, коли всі пластуни покинуть свої намети і підуть купатися в річку, Марічка планує непомітно з'їсти трохи печива. Звісно, Марічка не хоче бути покараною і дуже боїться, щоб Зеник не помітив пропажу.

Марічка підглядела, скільки пачок печива є в рюкзаку Зеника. Також вона знає, скільки штук в кожній пачці. Марічці не терпиться дізнатися, скільки ж печива вона зможе з'їсти так, щоб Зеник не помітив. Зеник помітить пропажу печива з деякої пачки тоді і тільки тоді, коли Марічка повністю спустошить її.

Вхідні дані

У першому рядку задано одне натуральне число n — кількість пачок печива.

У другому рядку задано np натуральних чисел a_i — кількість штук печива в i -й пачці.

Вихідні дані

У єдиному рядку виведіть одне ціле число — максимальну кількість штук печива, яку зможе з'їсти Марічка так, щоб Зеник не помітив цього.

Обмеження

20% тестів:

$$1 \leq n \leq 1000, 0 \leq a_i \leq 10^4$$

60% тестів:

$$1 \leq n \leq 10^5, 0 \leq a_i \leq 10^4$$

20% тестів:

$$1 \leq n \leq 10^5, 0 \leq a_i \leq 10^9$$

Algotester Task – Task 7

- Деталі завдання

Борщ, картопля і салат

Сьогодні в пластунів свято! Привезли вдвічі більше продуктів на обід. Зеник разом із Марічкою зголосилися допомагати на кухні. Всього є три різні страви: борщ, картопля і салат.

У Пласті в Зеника та Марічки є n друзів. i -й друг хоче з'їсти a_i грамів борщу, b_i грамів картоплі та c_i грамів салату. Якщо друг Зеника та Марічки хоче з'їсти xx грамів певної страви, а йому в тарілку поклали y грамів, то ступінь недовіри до Зеника з Марічкою в цього друга зростає на $|x - y|$ одиниць. Зенику з Марічкою наказали накладати всім однакові порції певної страви. Тобто кожному другу потрібно накласти A грамів борщу, B грамів картоплі та C грамів салату. Допоможіть Зенику та Марічці обрати такі A , B та C , щоб сумарний ступінь недовіри всіх друзів був якомога меншим.

Вхідні дані

У першому рядку задано одне ціле число n — кількість друзів Зеника та Марічки.

У наступних n рядках задано по три цілі числа a_i , b_i , c_i — кількість борщу, картоплі та салату, яку хоче отримати i -ий друг.

Вихідні дані

У єдиному рядку виведіть одне ціле число — сумарний ступінь недовіри всіх друзів до Зеника та Марічки.

Обмеження

$2 \leq n \leq 105$, n - парне.

30% тестів:

$0 \leq a_i, b_i, c_i \leq 444$

70% тестів:

$0 \leq a_i, b_i, c_i \leq 104$

Algotester Task – Task 8

- Деталі завдання

Lab 2v1

У вас є дорога, яка виглядає як N чисел.

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

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?

Вхідні дані

У першому рядку ціле число N - кількість чисел

У другому рядку масив r , який складається з NN цілих чисел

Вихідні дані

Єдине ціле число m - мінімальна втома, яку можна отримати

Обмеження

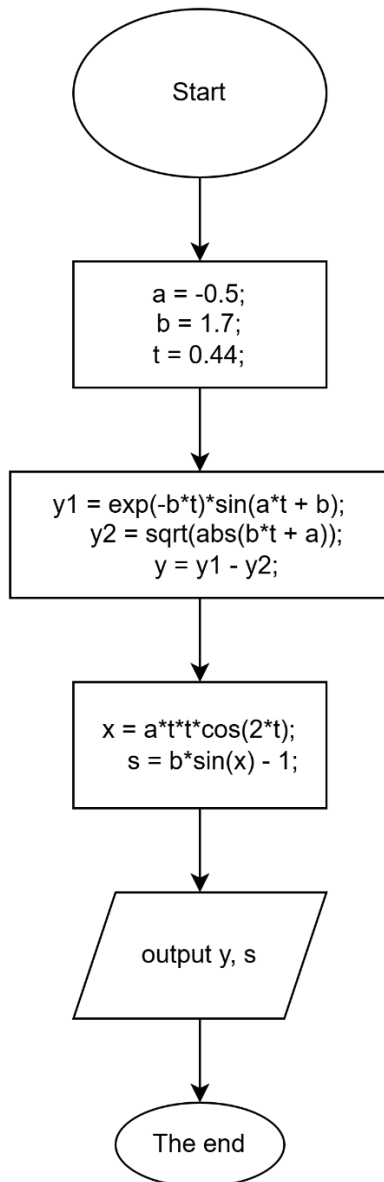
$1 \leq N \leq 10^5$

$0 \leq r_i \leq 10^5$

2. Дизайн та планована оцінка часу виконання завдань:

VNS Practice Work – Task 1

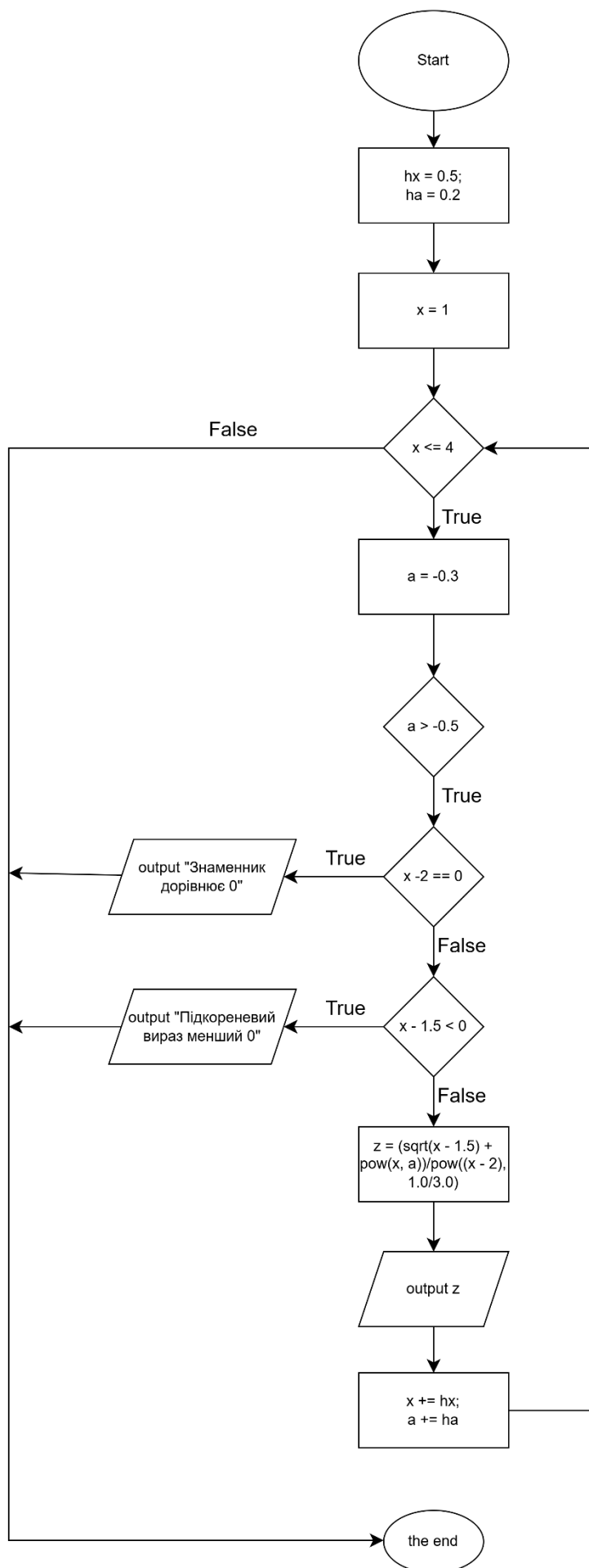
- Блок-схема



- Планований час на реалізацію 15 хв

VNS Practice Work – Task 2

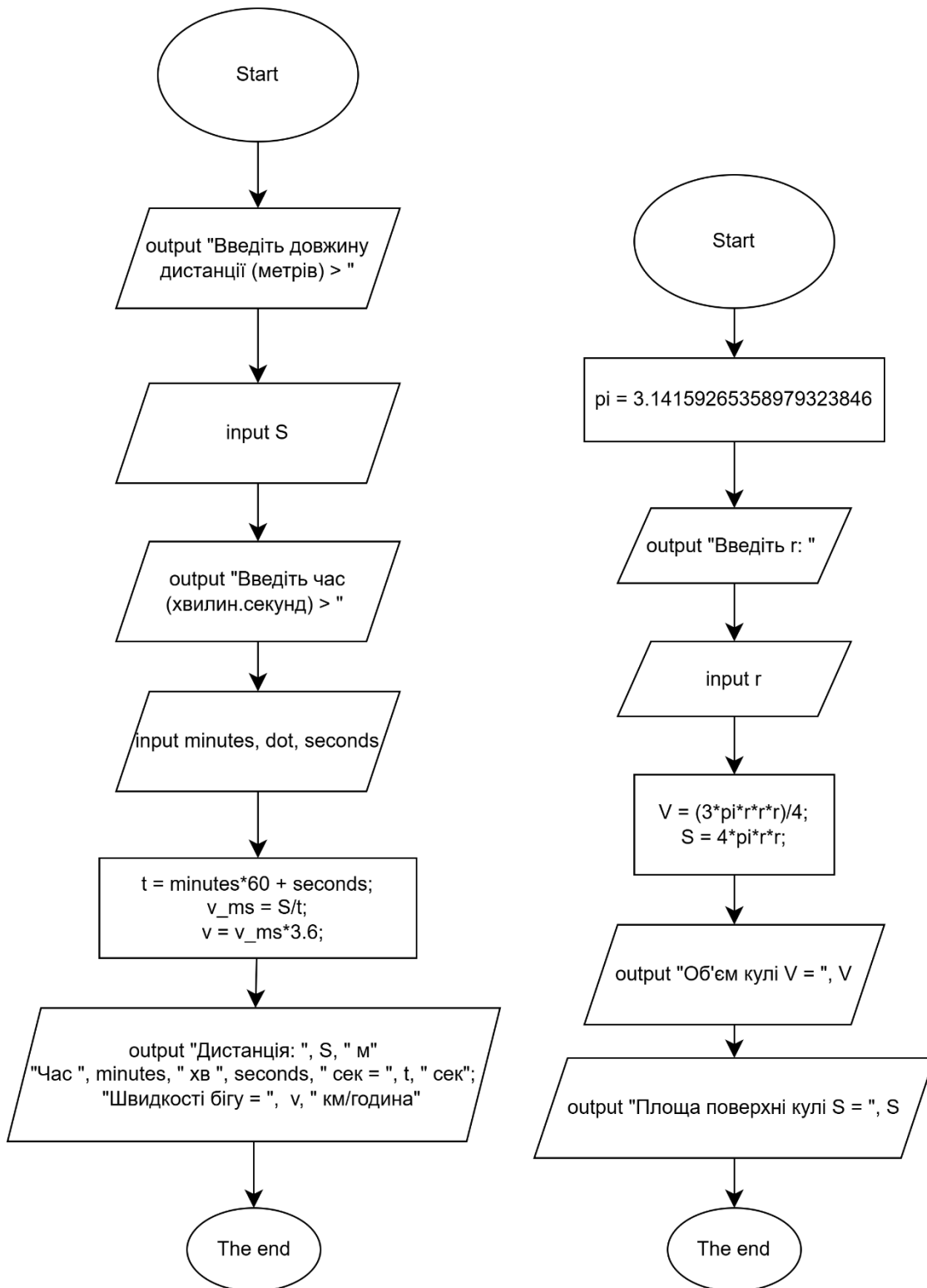
- Блок-схема



- Планований час на реалізацію 20 хв

VNS Practice Work – Task 3

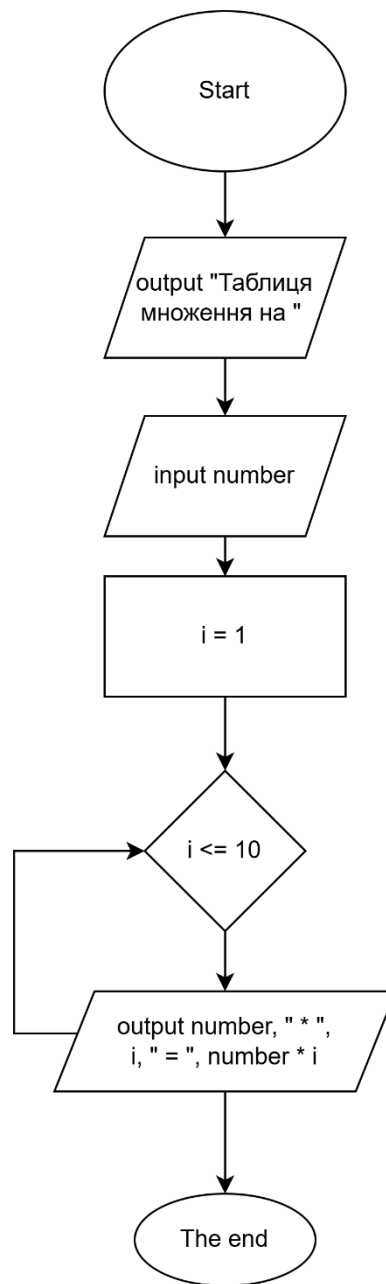
- Блок-схема



- Планований час на реалізацію 30 хв

VNS Practice Work – Task 4

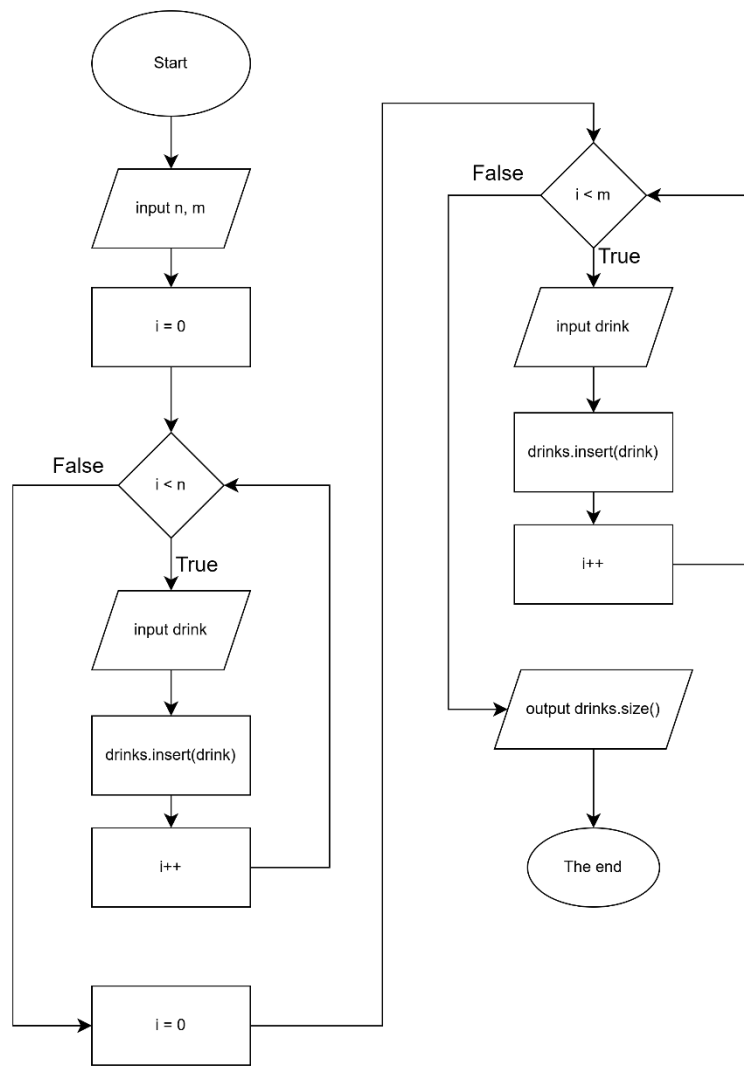
- Блок-схема



- Планований час на реалізацію 10 хв

Algotester Task – Task 5

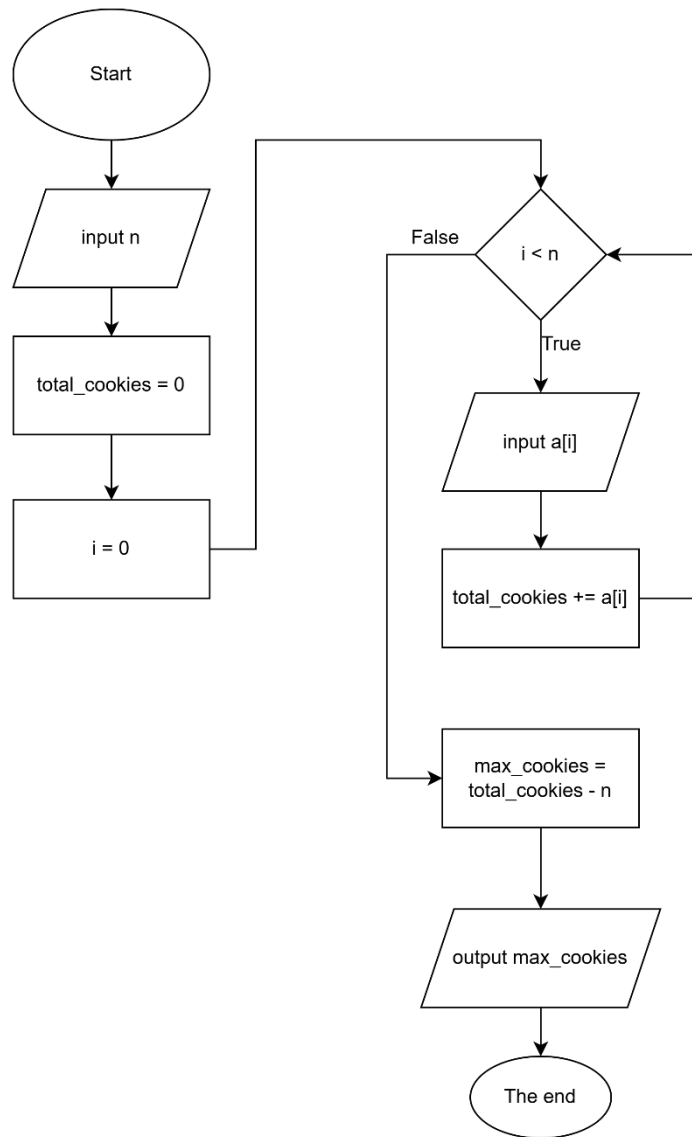
- Блок-схема



- Планований час на реалізацію 25 хв

Algotester Task – Task 6

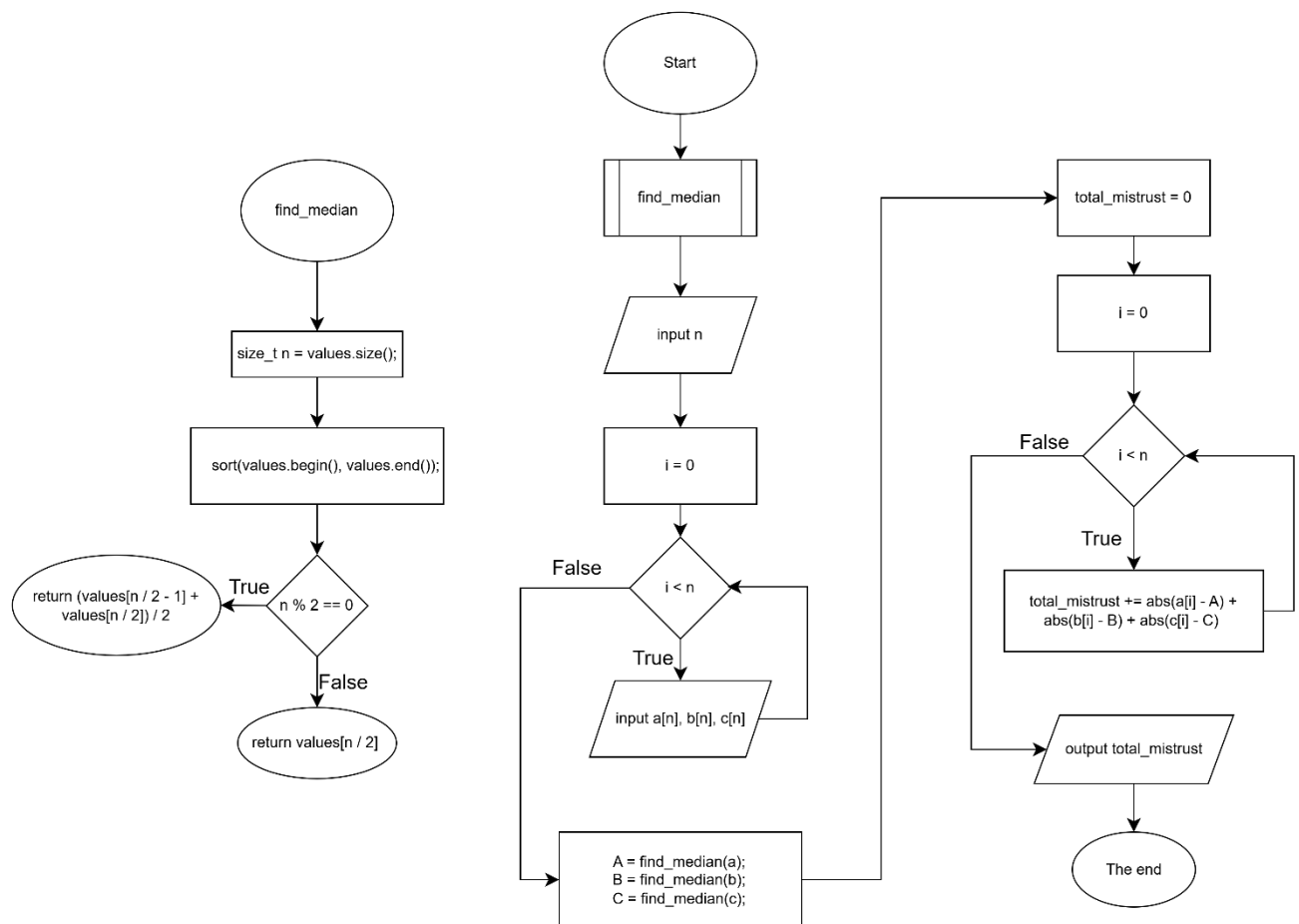
- Блок-схема



- Планований час на реалізацію 25 хв

Algotester Task – Task 7

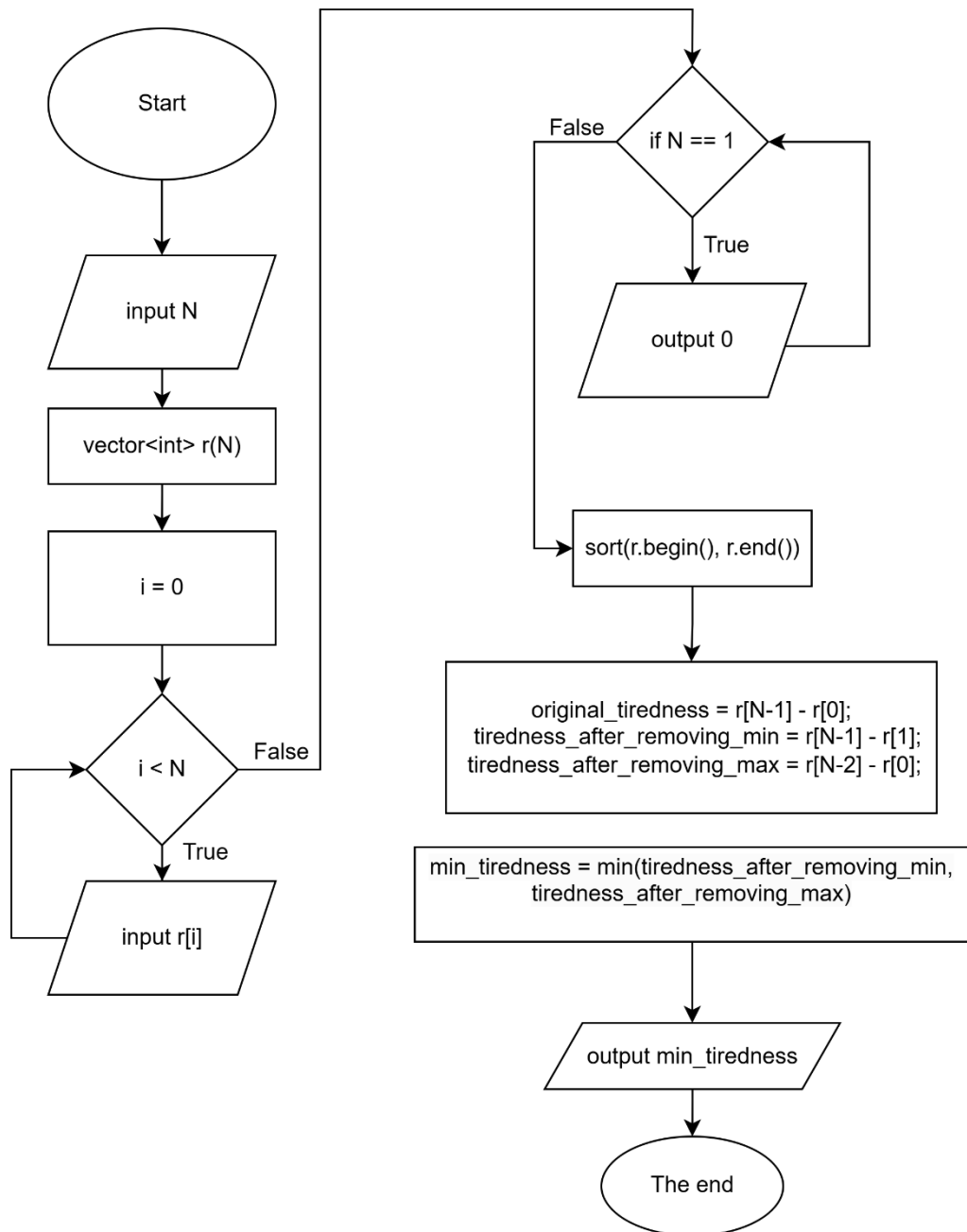
- Блок-схема



- Планований час на реалізацію 25 хв

Algotester Task – Task 8

- Блок-схема



- Планований час на реалізацію 25 хв

4. Код програм

VNS Practice Work – Task 1

```

#include <iostream>
#include <cmath>

using namespace std;

int main() {
    double a = -0.5;

```

```

double b = 1.7;
double t = 0.44;

double y1 = exp(-b*t)*sin(a*t + b);
double y2 = sqrt(abs(b*t + a));
double y = y1 - y2;

double x = a*t*t*cos(2*t);
double s = b*sin(x) - 1;

cout << y << endl;
cout << s << endl;

return 0;
}

```

VNS Practice Work – Task 2

```

#include <iostream>
#include <cmath>

using namespace std;

int main() {
    double x, a;

    double hx = 0.5;
    double ha = 0.2;

    for (x = 1; x <= 4; x += hx) {
        for (a = -0.3; a > -0.5; a += ha) {
            if (x - 2 == 0) {
                cout << "Знаменник дорівнює 0" << endl;
            }
            else if (x - 1.5 < 0) {
                cout << "Підкореневий вираз менший 0" << endl;
            }
            else {
                double z = (sqrt(x - 1.5) + pow(x, a))/pow((x - 2), 1.0/3.0);
                cout << "z = " << z << endl;
            }
        }
    }
}

```

```
    return 0;
}
```

VNS Practice Work – Task 3

```
#include <iostream>
#include <iomanip>

using namespace std;

int main() {
    double S;
    cout << "Введіть довжину дистанції (метрів) > ";
    cin >> S;

    int minutes;
    int seconds;
    cout << "Введіть час (хвилин.секунд) > ";
    char dot;
    cin >> minutes >> dot >> seconds;

    double t = minutes*60 + seconds;

    double v_ms = S/t;
    double v = v_ms*3.6;

    cout << "Дистанція: " << S << " м" << endl;
    cout << "Час " << minutes << " хв " << seconds << " сек = " << t << " сек" <<
endl;
    cout << "Швидкості бігу = " << fixed << setprecision(2) << v << " км/година" <<
endl;

    return 0;
}
```

Частина 2:

```
#include <iostream>
#include <iomanip>

using namespace std;

int main() {
    const double pi = 3.14159265358979323846;

    double r;
```

```

    cout << "Введіть r: ";
    cin >> r;

    double V = (3*pi*r*r*r)/4;
    double S = 4*pi*r*r;

    cout << "Об'єм кулі V = " << fixed << setprecision(2) << V << endl;
    cout << "Площа поверхні кулі S = " << fixed << setprecision(2) << S << endl;

    return 0;
}

```

VNS Practice Work – Task 4

```

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

int main() {
    int number;
    cout << "Таблиця множення на ";
    cin >> number;
    for (int i = 1; i <= 10; i++) {
        cout << number << " * " << i << " = " << number * i << endl;
    }
    return 0;
}

```

Algotester Task – Task 5

```

#include <iostream>
#include <set>
#include <string>
using namespace std;

int main() {
    int n, m;
    cin >> n >> m;

    set<string> drinks;

    for (int i = 0; i < n; i++) {
        string drink;
        cin >> drink;
        drinks.insert(drink);
    }

    for (int i = 0; i < m; i++) {

```



```

        string drink;
        cin >> drink;
        drinks.insert(drink);
    }

    cout << drinks.size() << endl;

    return 0;
}

```

Algotester Task – Task 6

```

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

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

    vector<long long> a(n);
    long long total_cookies = 0;

    for (int i = 0; i < n; ++i) {
        cin >> a[i];
        total_cookies += a[i];
    }

    long long max_cookies = total_cookies - n;

    cout << max_cookies << endl;
    return 0;
}

```

Algotester Task – Task 7

```

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

long long find_median(vector<int>& values) {
    size_t n = values.size();
    sort(values.begin(), values.end());
    if (n % 2 == 0) {
        return (values[n / 2 - 1] + values[n / 2]) / 2;
    }
}

```

```

    } else {
        return values[n / 2];
    }
}

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

    vector<int> a(n), b(n), c(n);

    for (int i = 0; i < n; ++i) {
        cin >> a[i] >> b[i] >> c[i];
    }

    long long A = find_median(a);
    long long B = find_median(b);
    long long C = find_median(c);

    long long total_mistrust = 0;

    for (int i = 0; i < n; ++i) {
        total_mistrust += abs(a[i] - A) + abs(b[i] - B) + abs(c[i] - C);
    }

    cout << total_mistrust << endl;

    return 0;
}

```

Algotester Task – Task 8

```

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

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

    vector<int> r(N);
    for (int i = 0; i < N; ++i) {
        cin >> r[i];
    }

    if (N == 1) {
        cout << 0 << endl;
    }
}

```

```

        return 0;
    }

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

    // Втома без видалення
    int original_tiredness = r[N-1] - r[0];

    // Втома після видалення мінімального елемента
    int tiredness_after_removing_min = r[N-1] - r[1];

    // Втома після видалення максимального елемента
    int tiredness_after_removing_max = r[N-2] - r[0];

    // Мінімальна втома
    int min_tiredness = min(tiredness_after_removing_min,
tiredness_after_removing_max);

    cout << min_tiredness << endl;

    return 0;
}

```

5. Результати виконання завдань, тестування та фактично затрачений час:

VNS Practice Work – Task 1

```

ebugAdapters\bin\WindowsDebugLauncher.exe' '--stdin=Microsoft-MIEngine-In-f11d4teg.zer
rosoft-MIEngine-Error-kzj4ovhd.tzo' '--pid=Microsoft-MIEngine-Pid-q0pgwdej.3hb' '--dbg
-0.0266334
-1.10478
PS D:\AI\ai_programming_playground_2024\ai_13\yelyzaveta_osinna\saga_1>

```

VNS Practice Work – Task 2

```

ebugAdapters\bin\WindowsDebugLauncher.exe' '--stdin=Microsoft-MIEngine-In-uymVCjn
rosoft-MIEngine-Error-adxl24oe.o2l' '--pid=Microsoft-MIEngine-Pid-q5qb2lfa.cfj' '
x = 1, a = -0.3: Підкореневий вираз менший 0
x = 1.5, a = -0.1: z = -1.20986
x = 2, a = 0.1: Знаменник дорівнює 0
x = 2.5, a = 0.3: z = 2.91846
x = 3, a = 0.5: z = 2.9568
x = 3.5, a = 0.7: z = 3.3351
x = 4, a = 0.9: z = 4.01878
PS D:\AI\ai_programming_playground_2024\ai_13\yelyzaveta_osinna\saga_1>

```

VNS Practice Work – Task 3

```

Microsoft-MIEngine-Error-gg4tfcmr.4l4' '--pid=Microsoft-MIEngine-Pid-c151hmcx
Введіть довжину дистанції (метрів) > 5000
Введіть час (хвилин.секунд) > 250.36
Дистанція: 5000 м
Час 250 хв 36 сек = 15036 сек
Швидкості бігу = 1.20 км/година
PS D:\AI\ai_programming_playground_2024\ai_13\yelyzaveta_osinna\saga_1>

```

```

ebugAdapters\bin\WindowsDebugLauncher.exe' '--stdin=Microsoft-MIEngine-In-xui
Microsoft-MIEngine-Error-44vjn4tq.l4p' '--pid=Microsoft-MIEngine-Pid-rfeleiww.fdv
Введіть r: 20
Об'єм кулі V = 18849.56
Площа поверхні кулі S = 5026.55
PS D:\AI\ai_programming_playground_2024\ai_13\yelyzaveta_osinna\saga_1>

```

VNS Practice Work – Task 4

```

ebugAdapters\bin\WindowsDebugLauncher.exe' '--stdin=Microsoft-MIEngine-In-inyg
Microsoft-MIEngine-Error-iqpd2qzg.ley' '--pid=Microsoft-MIEngine-Pid-otn0f22j.tqa
Таблиця множення на 7
7 * 1 = 7
7 * 2 = 14
7 * 3 = 21
7 * 4 = 28
7 * 5 = 35
7 * 6 = 42
7 * 7 = 49
7 * 8 = 56
7 * 9 = 63
7 * 10 = 70
PS D:\AI\ai_programming_playground_2024\ai_13\yelyzaveta_osinna\saga_1>

```

Algotester Task – Task 5

```

Microsoft-MIEngine-Error-45dbe5py.ne3' '--pid=Microsoft-MIEngine-Pid-orcp33ev.3a
1 2
fanta
pepsi
fanta
2
PS D:\AI\ai_programming_playground_2024\ai_13\yelyzaveta_osinna\saga_1>

```

Algotester Task – Task 6

```
Microsoft-MIEngine-Error-wrdxwjyl.jvj' '--pid=Microsoft-MIEngine-Pid-wqbrqrgc.pwf' '--  
4  
4 7 47 74  
128  
PS D:\AI\ai_programming_playground_2024\ai_13\yelyzaveta_osinna\saga_1>
```

Algotester Task – Task 7

```
Microsoft-MIEngine-Error-1drassor.cin' '--pid=Microsoft-MIEngine-Pid-mcizikar.jvd' '--u  
2  
10 50 100  
70 10 50  
150  
PS D:\AI\ai_programming_playground_2024\ai_13\yelyzaveta_osinna\saga_1>
```

Algotester Task – Task 8

```
debugAdapters\bin\windowsDebugLauncher.exe' '--stdin=Microsoft-MIEngine-In-p  
rosoft-MIEngine-Error-xjf5kw5e.qud' '--pid=Microsoft-MIEngine-Pid-u4t4sjkr.  
5  
1 2 2 4 4  
2  
PS D:\AI\ai_programming_playground_2024\ai_13\yelyzaveta_osinna\saga_1>
```

Час затрачений на виконання кожного завдання – 30 хв

Висновки:

Виконавши розрахункову роботу я на практиці застосувала знання, здобуті за семестр