

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-12
Лазаревич Юлія Дмитрівна

Львів 2024

Тема роботи:

Перевірка знань, які були здобуті за три місяці вивчення дисципліни «Основи програмування» з таких тем, як: функції вводу та виводу, цикли, умовні оператори, масиви, змінні та константи, створення та виклик функцій, їх перенавантаження, рекурсивні функції, використання структур та виконання різних операцій.

Мета роботи:

Перевірити, наскільки я володію поданими за цей час темами на практиці.

Виконання роботи:

1. Опрацювання завдання та вимог до програм та середовища:

Завдання №1 – VNS Practice Work – Task 1 – Variant 10.

Розробити лінійний алгоритм для розв'язання задачі.

$$a = \frac{\sqrt{|x+1|} - \sqrt{y}}{1 + \frac{x^2}{2} + \frac{y^2}{4}}; b = ctge^{x+3}. \text{Значення } x, y, z \text{ вибрати самостійно.}$$

Завдання №2 – VNS Practice Work – Task 2 – Variant 21.

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

Увести п'ять наборів сторін трикутника a, b, c і визначити, для яких сторін висота h_a буде найбільшою. Підказка: $h_a = \frac{2}{a} \sqrt{p(p-a)(p-c)}$, де $p = \frac{a+b+c}{2}$

Завдання №3 – VNS Practice Work – Task 3 – Variant 16.

Обчислення величини доходу по внеску.

Процентна ставка (% річних) і час зберігання (днів) задаються під час роботи програми. Нижче приведений вид екрану під час виконання програми, що рекомендується (дані, введені користувачем, виділені напівжирним шрифтом).

Обчислення величини доходу по внеску.

Введіть початкові дані:

Величина внеску (грн.) > **2500**

Термін внеску (днів) > 30

Процентна ставка (річних в %) > **20**

Дохід: 41.10 грн.

Сума, після закінчення терміну внеску: 2541.10 грн.

Обчислення об'єму порожнистого циліндра $V = \pi * h * (r_1^2 - r_2^2)$

де r_1 – радіус циліндра, r_2 – радіус отвору, h – висота циліндра.

Завдання №4 – VNS Practice Work – Task 4 – Variant 5.

Написати програму, яка обчислює суму перших n цілих позитивних чисел. Кількість підсумованих чисел повинна вводитися під час роботи програми.

Завдання №5 – Algotester Task – Task 5 – Робот.

Недавно батьки подарували Петрикові робота, якого можна програмувати. Початково робот стоїть у точці з координатами (0, 0). Петрик уводить роботу набір команд, які той виконує послідовно від першої до останньої. Існує два типи команд:

1. **U** — перейти на 1 вгору, тобто з точки (x, y) у точку $(x, y+1)$.
2. **R** — перейти на 1 вправо, тобто з точки (x, y) у точку $(x+1, y)$.

Петрик запрограмував робота послідовністю команд s_1, s_2, \dots, s_n . Вам задана ця послідовність, а також пара чисел x та y .

Визначте, чи може Петрик переставити місцями команди так, щоб робот пройшов через точку (x, y) .

Вхідні дані

У першому рядку задано рядок s , який складається з n символів, які відповідають за команди, введені Петриком.

У другому рядку задано два цілих числа x та y .

Вихідні дані

У єдиному рядку виведіть **YES**, якщо Петрик може переставити місцями команди так, щоб робот перейшов через точку (x, y) .

Якщо досягнути цілі неможливо, виведіть **NO**.

Завдання №6 – Algotester Task – Task 6 – Менші зліва, менші справа.

Організатори змагань загадали масив a_i з n елементів.

Для кожного елемента відомо значення l_i — кількість менших елементів зліва від нього та r_i — кількість менших елементів справа. Знайдіть будь-який масив, який підходить під дані обмеження.

Для заданих вхідних даних гарантовано існує хоча б один масив. Якщо існує декілька масивів, які задовольняють умови, виведіть будь-який з них.

Вхідні дані:

У першому рядку задано єдине ціле число n — довжину масиву.

У наступних n рядках задано по 2 цілих числа — l_i та r_i .

Вихідні дані:

У єдиному рядку виведіть n чисел — значення елементів масиву. Усі значення повинні бути в межах від 0 до 109.

Якщо існує декілька масивів, які задовольняють умови, виведіть будь-який з них.

Завдання №7 – Algotester Task – Task 7 – Допоможе чи заб'є.

До Тойлет-мена, відомого вам білоруського супергероя, дуже часто звертаються по допомогу різні люди. Проте, очевидно, усім мужній гігант допомогти не зможе — не вистачить часу. Та й не дуже хоче.

Саме тому, коли до нього приходить певне SMS-повідомлення з проханням про допомогу, він погодиться допомогти тоді й лише тоді, коли в цьому повідомленні знайдеться хоча б k входжень рядка **TOILET**, які не перетинаються.

За заданим повідомленням s , яке складається з великих латинських символів, виведіть **YES**, якщо Тойлет-мен погодиться допомогти людині, яка написала це повідомлення. У протилежному разі виведіть **NO**.

Вхідні дані

У першому рядку задано одне ціле число k — мінімальна кількість незалежних входжень рядка **TOILET**.

Другий рядок містить рядок s — повідомлення, надіслане Тойлет-мену.

Рядок містить лише великі латинські символи.

Вихідні дані

У єдиному рядку виведіть **YES** або **NO** — відповідь на задачу.

Завдання №8 – Algotester Task – Task 8 – Lab 1 – V1.

У вашого персонажа є N хітпойнтів та M мани.

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

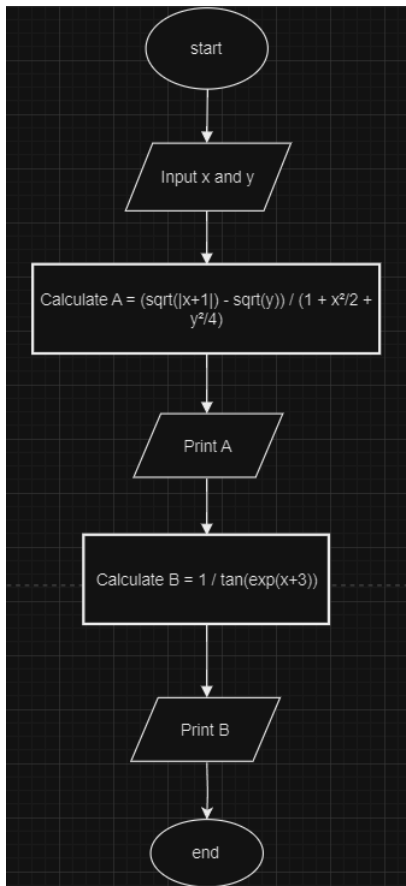
Якщо якийсь закляття забирає і хітпойнти і ману - ваш персонаж програє, отже для виграшу треба використовувати при одному заклинанні **АБО** хітпойнти, **АБО** ману.

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

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

2. Код програм з блок схемою та посиланням на зовнішні ресурси:

Завдання №1 – VNS Practice Work - Task 1.



```
vns_practice_work_1_task_1_yuliia_lazarevych.cpp > main()
1 //a=(sqrt(|x+1|) - sqrt(y))/(1+x^2/2+y^2/4); b=ctge^(x+3).
2
3 #include <iostream>
4 #include <cmath>
5 using namespace std;
6
7 int main(){
8     int x, y;
9     cin >> x >> y;
10    long double a, a1h, a2h, a3l;
11    a1h=sqrt(fabs(x + 1));
12    a2h=sqrt(y);
13    a3l=1 + (pow(x,2)/2) + (pow(y,2)/4);
14    a=(a1h - a2h) / a3l;
15    cout << a << endl;
16
17    long double b, b1, b2;
18    b2=exp(x+3); //e
19    b=1/tan(b2);
20    cout << b << endl;
21    return 0;
22 }
```

```
1
2
0
0.39909
```

ai_programming_playground_2024/ai_12/yuliia_lazarevych/saga_1/codes/vns_practice_work_1_task_1_yuliia_lazarevych.cpp at saga_1_practice_work_yuliia_lazarevych · artificial-intelligence-department/ai_programming_playground_2024

Завдання №2 – VNS Practice Work - Task 2.

```

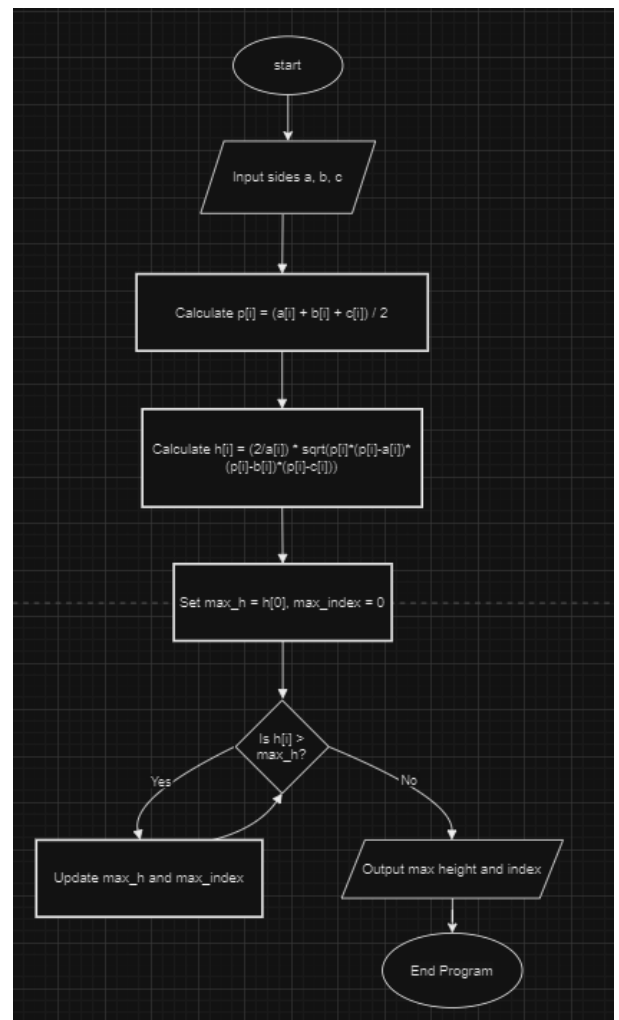
1 //Увести п'ять наборів сторін трикутника a, b, c і визначити, для яких сторін висота h_a буде найбільшою.
2 //Підказка:  $h_a = 2/a \sqrt{p(p-a)(p-b)(p-c)}$ , де  $p = (a+b+c)/2$ 
3
4 #include<iostream>
5 #include<cmath>
6 using namespace std;
7
8 int main(){
9     const int N = 5; //константа для збереження кількості трикутників
10    int a[N], b[N], c[N]; //одновимірні масиви для зберігання сторін трикутників
11
12    cout << "Введіть сторони a трикутників: " << endl; //оператори виведення
13    for (int i = 0; i < N; i++) {
14        cin >> a[i]; //оператори введення
15    }
16
17    cout << "Введіть сторони b трикутників: " << endl;
18    for (int i = 0; i < N; i++) {
19        cin >> b[i];
20    }
21
22    cout << "Введіть сторони c трикутників: " << endl;
23    for (int i = 0; i < N; i++) {
24        cin >> c[i];
25    }
26
27    double p[N], h[N]; //масиви для напівпериметрів і висот
28    for (int i = 0; i < N; i++) {
29        p[i] = (a[i] + b[i] + c[i]) / 2.0; // Обчислення напівпериметра
30        h[i] = (2.0 / a[i]) * sqrt(p[i] * (p[i] - a[i]) * (p[i] - b[i]) * (p[i] - c[i])); //математичні операції для виконання мат. обчислень
31    }
32
33    double max_h = h[0]; //дійсна змінна для подвійної точності обчислень
34    int max_index = 0;
35    for (int i = 1; i < N; i++) { //цикл для перевірки висот та знаходження максимальної
36        if (h[i] > max_h) { //умовне розгалуження для перевірки висот та знаходження максимальної
37            max_h = h[i];
38            max_index = i;
39        }
40    }
41
42    cout << "Найбільша висота: h" << (max_index + 1) << " = " << max_h << endl;
43    return 0;
44 }

```

```

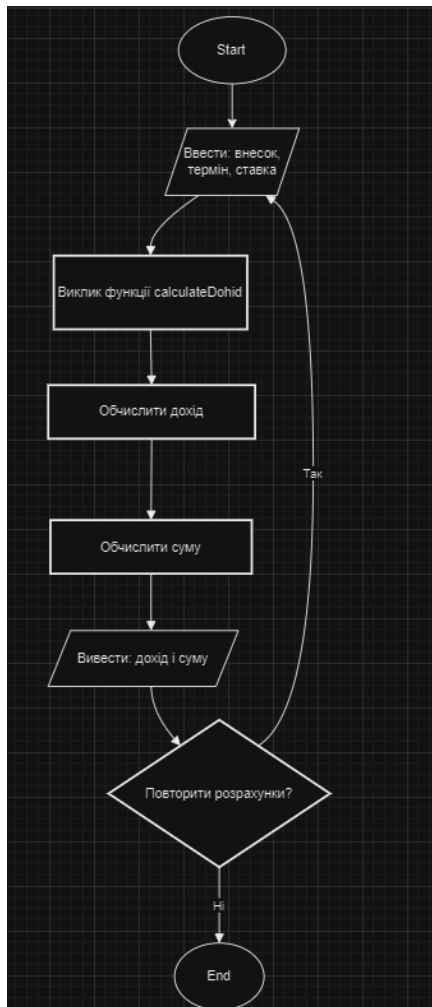
Введіть сторони a трикутників:
1
2
5
6
7
Введіть сторони b трикутників:
2
5
3
1
4
Введіть сторони c трикутників:
2
8
4
7
5
8
4
7
5
5
Найбільша висота: h5 = 2.79942

```



Завдання №3 – VNS Practice Work – Task 3 – Variant 16.

```
vns_practice_work_1_task_3_yuliia_lazarevych.cpp > ...
1 //Обчислення величини доходу по внеску.
2 //Процентна ставка (% річних) і час зберігання (днів) задаються під час роботи програми.
3
4 #include <iostream>
5 #include <cmath>
6
7 using namespace std;
8
9 double calculateDohid(int vnesok, double stavka, double termin) { //функція для обчислення доходу
10     return vnesok * (stavka/100.0) * (termin / 365.0); //мат. операції для обчислення доходу
11 }
12
13 int main(){
14     char repeat;
15     do{ //цикл `do while` для повторення розрахунків за бажанням юзера
16         int vnesok, termin, stavka;
17         double sum;
18         cout << "Обчислення величини доходу по внеску.\n" << "Введіть початкові дані:\n" << "Величина внеску (грн.) " << endl;
19         cin >> vnesok;
20         cout << "Термін внеску (днів) " << endl;
21         cin >> termin;
22         cout << "Процентна ставка (річних в %) " << endl;
23         cin >> stavka;
24
25         double dohid=calculateDohid(vnesok, stavka, termin);
26         sum = vnesok + dohid;
27         cout << "Дохід: " << dohid << endl << "Сума, після закінчення терміну внеску: " << sum << endl;
28         cout << "Бажаєте виконати ще один розрахунок?(y/n): ";
29         cin >> repeat;
30     } while (repeat == 'y' or repeat == 'Y');
31     return 0;
32 }
```

```

Обчислення величини доходу по внеску.
Введіть початкові дані:
Величина внеску (грн.)
2500
Термін внеску (днів)
30
Процентна ставка (річних в %)
20
Дохід: 41.0959
Сума, після закінчення терміну внеску: 2541.1
Бажаєте виконати ще один розрахунок?(y/n): y
Обчислення величини доходу по внеску.
Введіть початкові дані:
Величина внеску (грн.)
100
Термін внеску (днів)
20
Процентна ставка (річних в %)
10
Дохід: 0.547945
  
```

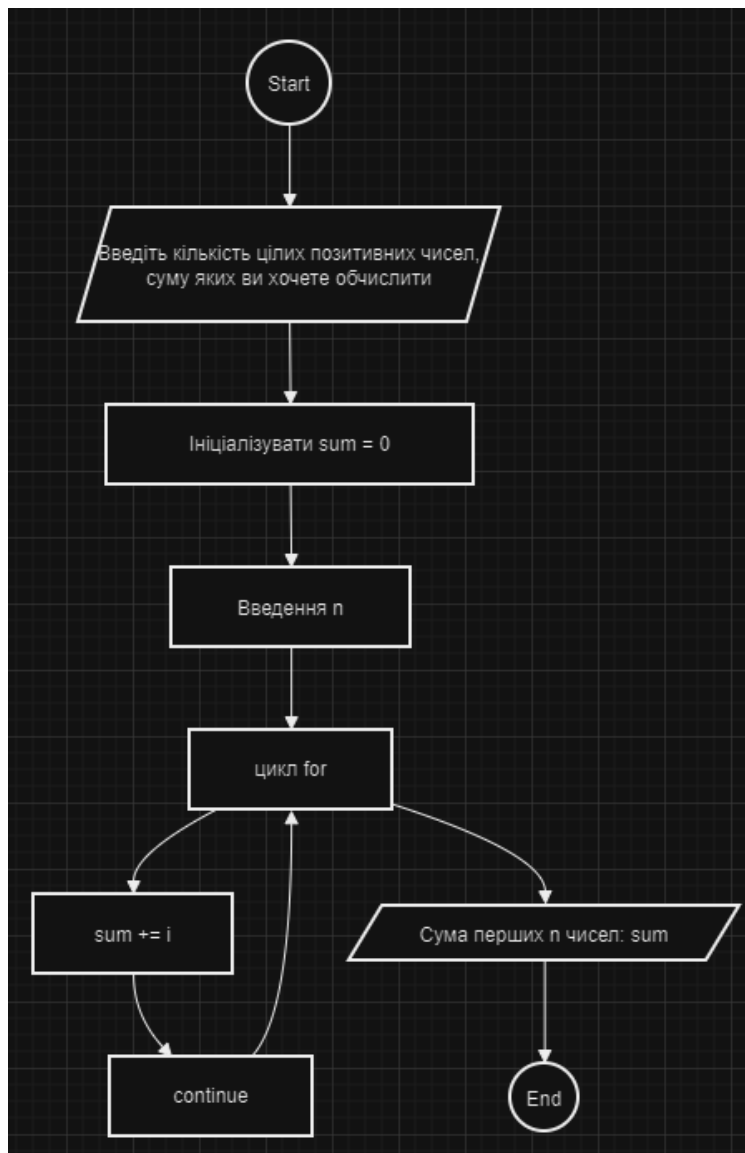
ai_programming_playground_2024/ai_12/yuliia_lazarevych/saga_1/codes/vns_practice_work_1_task_3_yuliia_lazarevych.cpp at [saga_1_practice_work_yuliia_lazarevych](#) · [artificial-intelligence-department/ai_programming_playground_2024](#)

Завдання №4 – VNS Practice Work – Task 4 – Variant 5.

```

vns_practice_work_1_task_4_yuliia_lazarevych.cpp > ...
1 //Написати програму, яка обчислює суму перших n цілих позитивних чисел.
2 //Кількість підсумованих чисел повинна вводитися під час роботи програми.
3
4 #include <iostream>
5 #include <cmath>
6
7 using namespace std;
8 int main(){
9     int n, sum=0;
10    cout << "Введіть кількість цілих позитивних чисел, суму яких ви хочете обчислити: ";
11    cin >> n;
12    for (int i=1; i<=n; ++i){//цикл фор для проходження по n-й кількості чисел
13        sum+=i;
14        continue;//оператор для продовження проходження циклу
15    }
16    cout << "Сума перших " << n << " чисел: " << sum << endl;
17    return 0;
18 }
  
```

Введіть кількість цілих позитивних чисел, суму яких ви хочете обчислити: 5
Сума перших 5 чисел: 15



[ai_programming_playground_2024/ai_12/yuliia_lazarevych/saga_1/codes/vns_practice_work_1_task_4_yuliia_lazarevych.cpp](https://github.com/yuliia-lazarevych/saga-1-codes/vns_practice_work_1_task_4_yuliia_lazarevych.cpp) at saga_1_practice_work_yuliia_lazarevych · artificial-intelligence-department/ai_programming_playground_2024

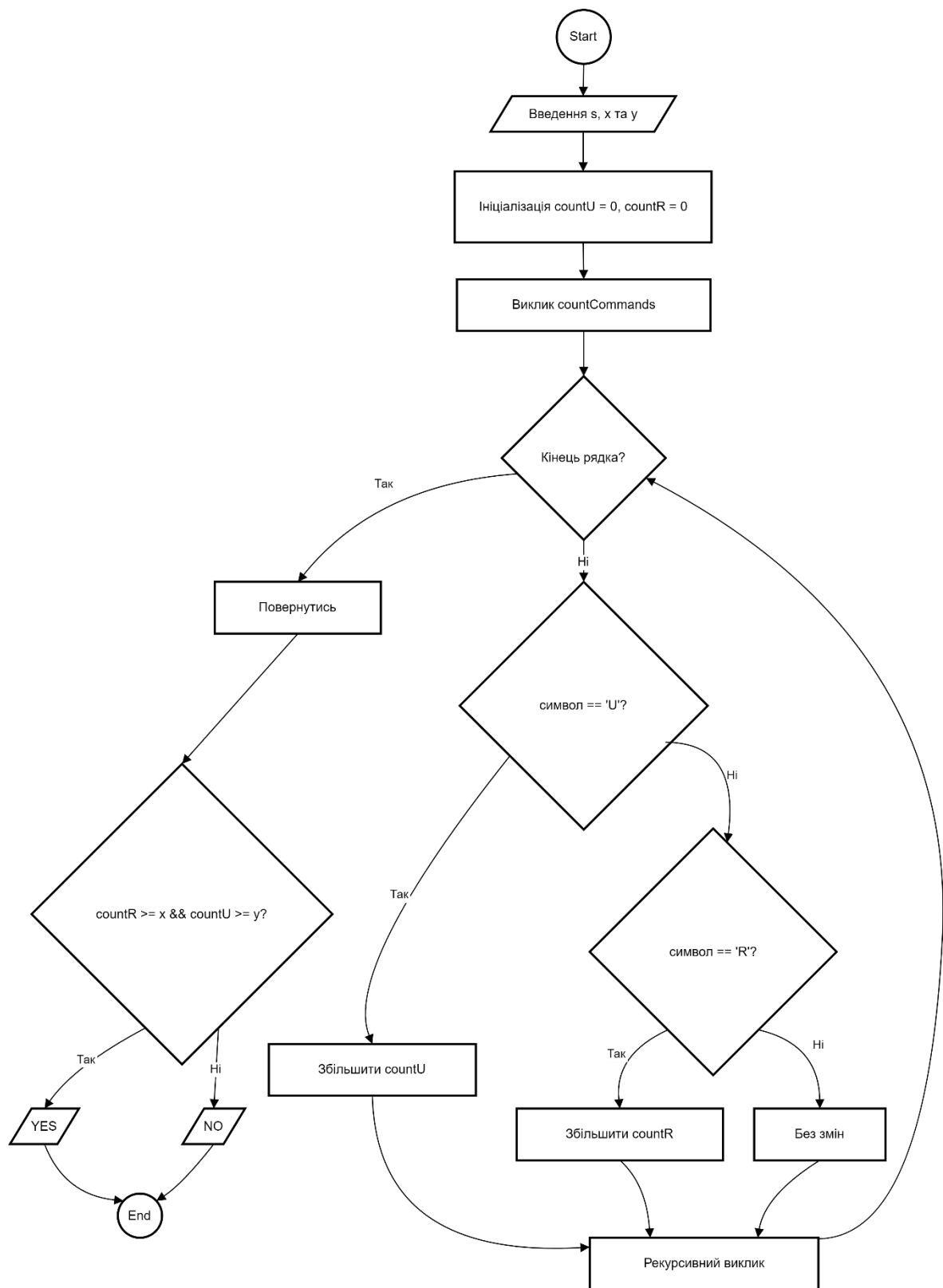
Завдання №5 – Algotester Task – Task 5 – [Робот](#).

```

1  //робот
2
3  #include <iostream>
4  #include <string>
5
6  using namespace std;
7
8  void countCommands(const string& s, int index, int& countU, int& countR) { //рекурсивна функція для підрахунку кількості команд
9      if (index == s.length()) return;
10     if (s[index] == 'U') countU++;
11     if (s[index] == 'R') countR++;
12
13     countCommands(s, index + 1, countU, countR); //рекурсивний виклик
14 }
15
16 int main() {
17     string s;
18     int x, y;
19     cin >> s >> x >> y;
20     int countU = 0, countR = 0;
21     countCommands(s, 0, countU, countR);
22     if (countR >= x && countU >= y) {
23         cout << "YES" << endl;
24     } else {
25         cout << "NO" << endl;
26     }
27
28     return 0;
29 }

```

Створено	Компілятор	Результат	Час (сек.)	Пам'ять (МіБ)	Дії
декілька секунд тому	C++ 23	Зараховано	0.003	1.172	Перегляд
хвилину тому	C++ 23	Зараховано	0.003	1.121	Перегляд

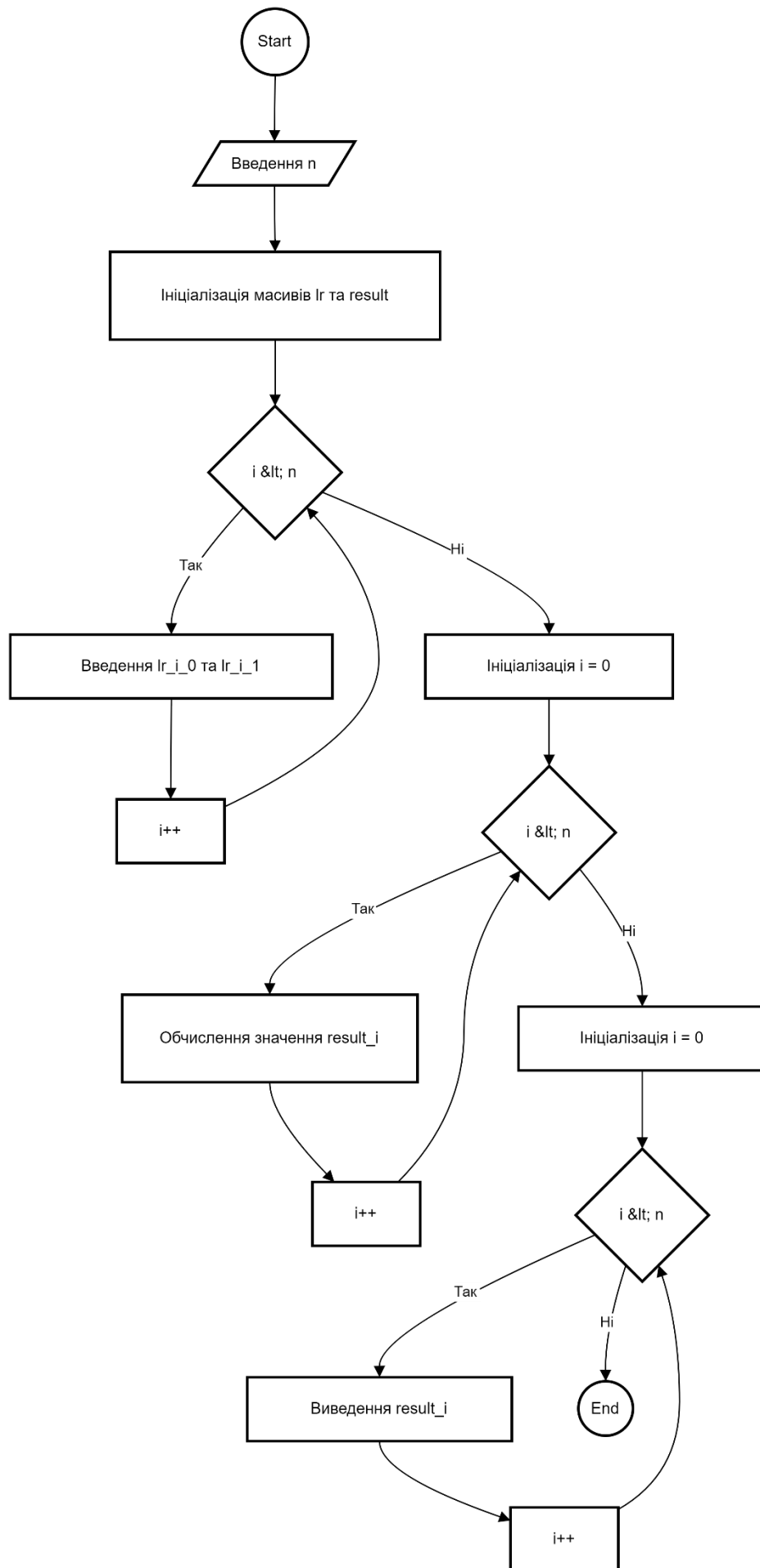


[ai programming playground 2024/ai 12/yuliia lazarevych/saga_1/codes/algotester practice work task 5 yuliia lazarevych.cpp](https://github.com/yulii-lazarevych/saga_1_codes_algotester_practice_work_task_5_yulii_lazarevych.cpp) at saga_1 practice work yuliia lazarevych · artificial-intelligence-department/ai programming playground 2024

Завдання №6 – Algotester Task – Task 6 – [Менші зліва, менші справа.](#)

```
algotester_practice_work_task_6_yuliia_lazarevych.cpp > ...
1  //Менші зліва менші справа
2
3  #include <iostream>
4  #include <vector>
5
6  using namespace std;
7
8  int main() {
9      int n, i = 0;
10     cin >> n;
11
12     vector<vector<int>> lr(n, vector<int>(2)); //двохвимірний масив для зберігання l та r
13     vector<int> result(n, 0);
14
15     while (i < n) { //використовую цикл while для зчитування l_i та r_i
16         cin >> lr[i][0] >> lr[i][1];
17         i++;
18     }
19
20     i = 0;
21     while (i < n) { //цикл while для обчислення значень елементів масиву
22         int count_left = lr[i][0];
23         int count_right = lr[i][1];
24         int value = count_left + count_right + 1;
25         result[i] = value;
26         i++;
27     }
28
29     i = 0;
30     while (i < n) {
31         cout << result[i] << " ";
32         i++;
33     }
34     cout << endl;
35
36     return 0;
37 }
```

Створено	Компілятор	Результат	Час (сек.)	Пам'ять (МіБ)	Дії
декілька секунд тому	C++ 23	Зараховано	0.086	7.359	Перегляд

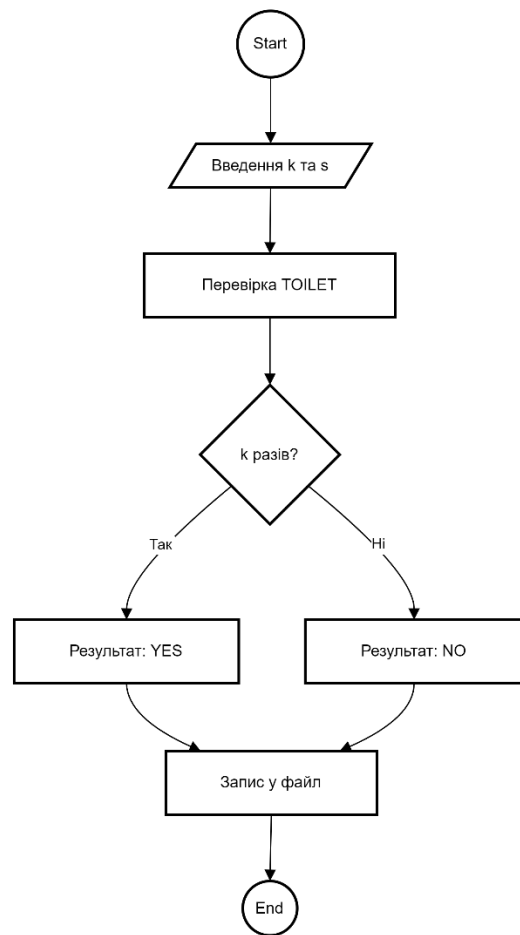


[ai_programming_playground_2024/ai_12/yuliia_lazarevych/saga_1/codes/algotester_practice_work_task_6_yuliia_lazarevych.cpp](#) at [saga_1_practice_work_yuliia_lazarevych · artificial-intelligence-department/ai_programming_playground_2024](#)

Завдання №7 – Algotester Task – Task 7 – Допоможе чи заб’є.

```
1 //допоможе чи заб'є
2
3 #include <iostream>
4 #include <fstream>
5 #include <string>
6 using namespace std;
7
8 void readInput(int &k, string &s) { //використовую переваження функцій для зчитування змінних
9     cin >> k >> s;
10 }
11
12 void readInputFromFile(const string &filename, int &k, string &s) {
13     ifstream inFile(filename);
14     inFile >> k >> s;
15 }
16
17 bool checkMessage(int k, const string &s) {
18     int count = 0;
19     size_t pos = 0;
20
21     while ((pos = s.find("TOILET", pos)) != string::npos) {
22         count++;
23         pos += 6;
24     }
25
26     return count >= k;
27 }
28
29 void writeResultToFile(const string &filename, const string &result) { //використовую функцію для запису результату у файл(робота з файлами)
30     ofstream outFile(filename);
31     outFile << result << endl;
32 }
33
34 int main() {
35     int k;
36     string s;
37
38     readInput(k, s);
39     string result = checkMessage(k, s) ? "YES" : "NO";
40     cout << result << endl;
41
42     writeResultToFile("output.txt", result); //записує результати у файл
43
44     return 0;
45 }
```

Створено	Компілятор	Результат	Час (сек.)	Пам'ять (МіБ)	Дії
декілька секунд тому	C++ 23	Зараховано	0.005	1.246	Перегляд



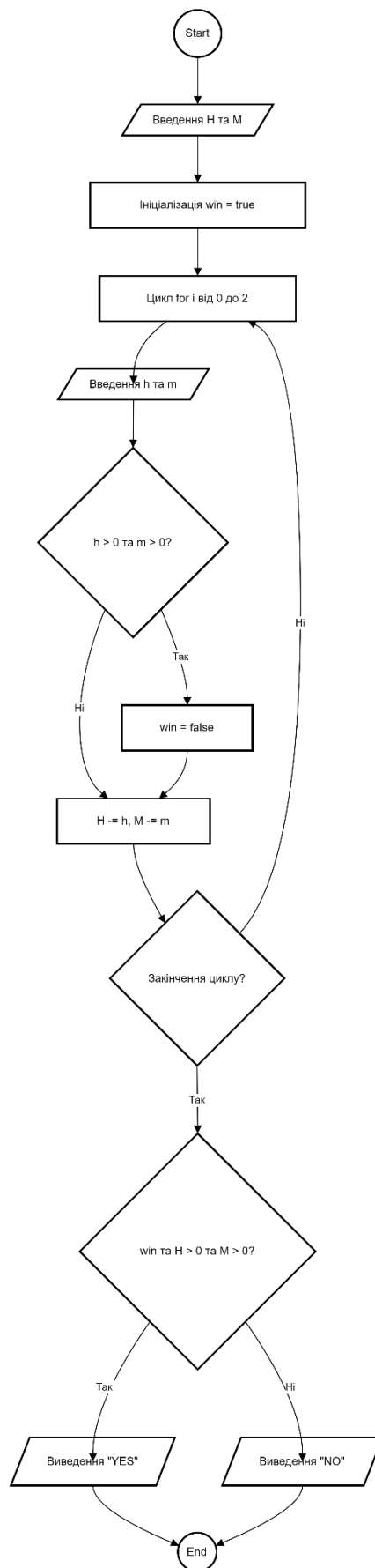
[ai_programming_playground_2024/ai_12/yuliia_lazarevych/saga_1/codes/algotester_practice_work_task_7_yuliia_lazarevych.cpp at saga_1_practice_work_yuliia_lazarevych · artificial-intelligence-department/ai_programming_playground_2024](#)

Завдання №8 – Algotester Task – Task 8 – Lab 1 – V1.


```

1  //лаб-1-1
2
3  #include <iostream>
4  using namespace std;
5
6  int main() {
7      long long H, M;
8      cin >> H >> M;
9      bool win = true;
10
11      for (int i = 0; i < 3; i++) {
12          long long h, m;
13          cin >> h >> m;
14
15          if (h > 0 && m > 0) {
16              win = false;
17          }
18
19          H -= h;
20          M -= m;
21      }
22
23      if (win && H > 0 && M > 0) {
24          cout << "YES" << endl;
25      } else {
26          cout << "NO" << endl;
27      }
28
29      return 0;
30  }

```



Створено	Компілятор	Результат	Час (сек.)	Пам'ять (МіБ)	Дії
декілька секунд тому	C++ 23	Зараховано	0.002	1.211	Перегляд

[ai programming playground 2024/ai 12/yuliia lazarevych/saga 1/codes/algotester practice work task 8_yuliia lazarevych.cpp at saga 1 practice work yuliia lazarevych · artificial-intelligence-department/ai programming playground 2024](#)