

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 ІІІ-12
Сирватка Олександр

Тема: Розрахункова робота.

Мета роботи: Одержати практичні навчки та закріпити знання, одержані протягом семестру.

Виконання роботи

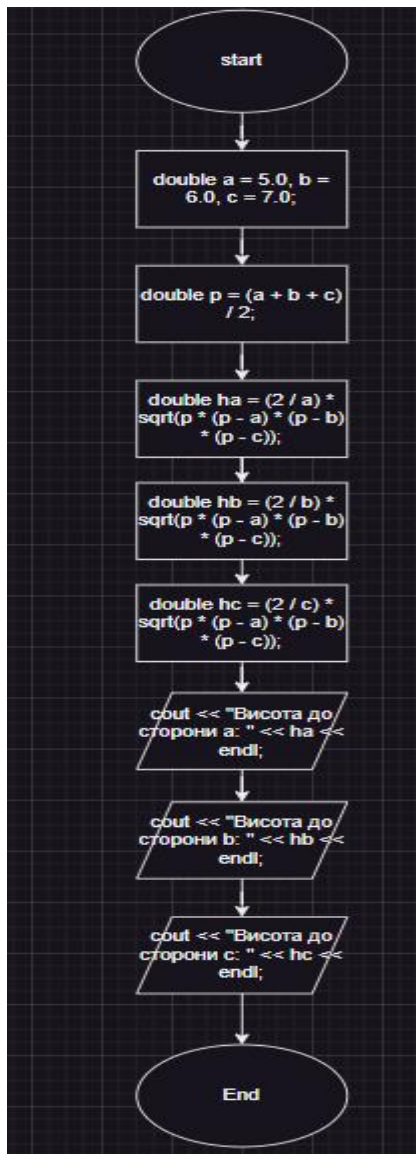
Завдання 1: VNS Practice Work Task 1

Розробити лінійний алгоритм для розв'язання задачі.

Варіант 1. h_a , h_b , h_c – висоти трикутника за формулою $h = \frac{2}{a} \sqrt{p(p-a)(p-b)(p-c)}$, де $p = (a+b+c)/2$. Значення сторін трикутника вибрати самостійно.

```
1  #include <iostream>
2  #include <cmath>
3  using namespace std;
4
5  int main() {
6      double a = 5.0, b = 6.0, c = 7.0;
7
8      double p = (a + b + c) / 2;
9
10     double ha = (2 / a) * sqrt(p * (p - a) * (p - b) * (p - c));
11     double hb = (2 / b) * sqrt(p * (p - a) * (p - b) * (p - c));
12     double hc = (2 / c) * sqrt(p * (p - a) * (p - b) * (p - c));
13
14     cout << "Висота до сторони a: " << ha << endl;
15     cout << "Висота до сторони b: " << hb << endl;
16     cout << "Висота до сторони c: " << hc << endl;
17
18     return 0;
19 }
20
```

```
Висота до сторони a: 5.87878
Висота до сторони b: 4.89898
Висота до сторони c: 4.19913
PS C:\Users\supes\projects>
```

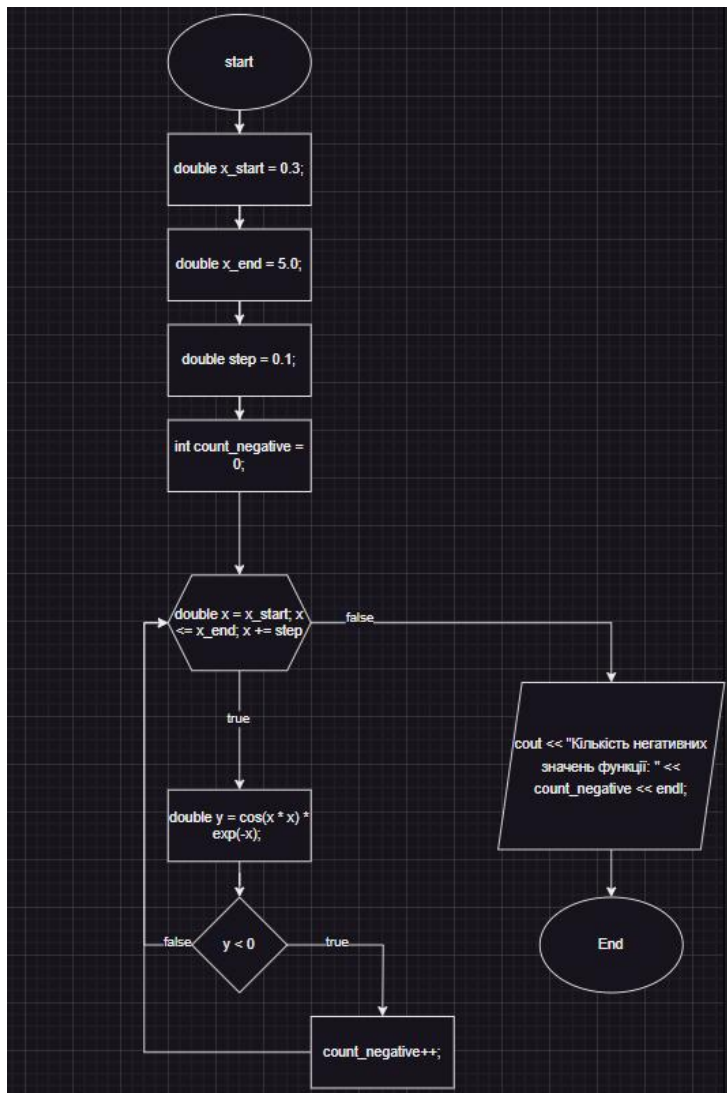


Завдання 2: VNS Practice Work Task 2

Розробити алгоритм, що розгалужується для розв'язання задачі.

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

```
1  #include <iostream>
2  #include <cmath>
3  using namespace std;
4
5  int main() {
6      // Параметри
7      double x_start = 0.3;
8      double x_end = 5.0;
9      double step = 0.1;
10     int count_negative = 0; // Лічильник негативних значень
11
12     // Цикл для перебору значень x у заданому діапазоні
13     for (double x = x_start; x <= x_end; x += step) {
14         double y = cos(x * x) * exp(-x); // Обчислення функції
15
16         // Перевірка на негативне значення
17         if (y < 0) {
18             count_negative++;
19         }
20     }
21
22     // Виведення результату
23     cout << "Кількість негативних значень функції: " << count_negative << endl;
24
25     return 0;
26 }
27
```



Кількість негативних значень функції: 21

Завдання 3: VNS Practice Work Task 3

Варіант 1. Обчислення об'єму паралелепіпеда. Нижче приведений вид екрану під час виконання програми, що рекомендується (дані, які вводяться користувачем, виділені напівжирним шрифтом).

Обчислення; об'єму паралелепіпеда.

Введіть початкові дані:

Довжина (см) > **9**

Ширина(см)> **7.5**

Висота(см)> **5**

Об'єм: 337.50 куб.см.

Реалізувати перерахунок ваги з фунтів в кілограми (один фунт - це 405,9 грами).

```

1  #include <iostream>
2  using namespace std;
3
4  int main() {
5      // Перше завдання: обчислення об'єму паралелепіпеда
6      double length, width, height, volume;
7
8      cout << "Обчислення об'єму паралелепіпеда." << endl;
9      cout << "Введіть початкові дані:" << endl;
10
11     // Введення довжини, ширини і висоти
12     cout << "Довжина (см) > ";
13     cin >> length;
14     cout << "Ширина (см) > ";
15     cin >> width;
16     cout << "Висота (см) > ";
17     cin >> height;
18
19     volume = length * width * height;
20
21     cout << "Об'єм: " << volume << " куб. см." << endl;
22
23     // Друге завдання: перерахунок ваги з фунтів у кілограми
24     double pounds, kilograms;
25
26     cout << "\nПерерахунок ваги з фунтів у кілограми." << endl;
27     cout << "Введіть вагу у фунтах > ";
28     cin >> pounds;
29
30     kilograms = pounds * 0.453592;
31
32     cout << "Вага у кілограмах: " << kilograms << " кг." << endl;
33
34     return 0;
35 }
36

```

```

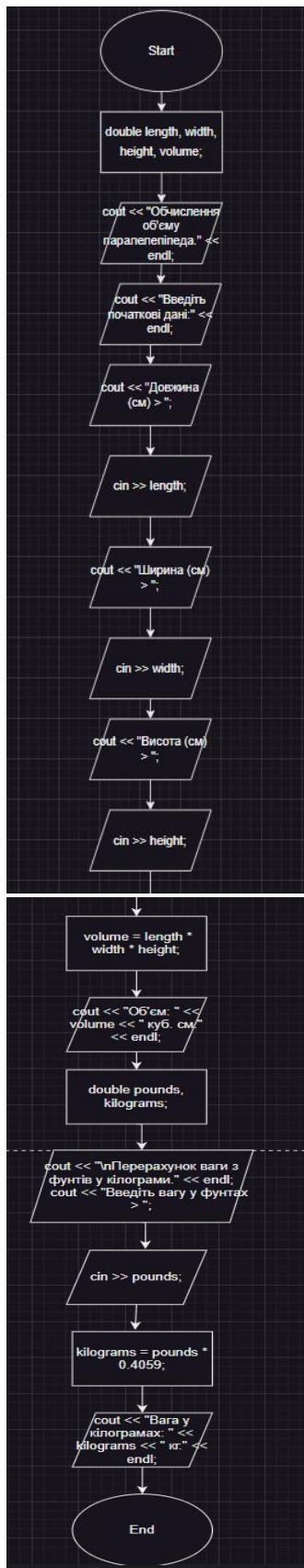
Обчислення об'єму паралелепіпеда.
Введіть початкові дані:
Довжина (см) > 9
Ширина (см) > 7.5
Висота (см) > 5
Об'єм: 337.5 куб. см.

```

```

Перерахунок ваги з фунтів у кілограми.
Введіть вагу у фунтах > 50
Вага у кілограмах: 22.6796 кг.

```



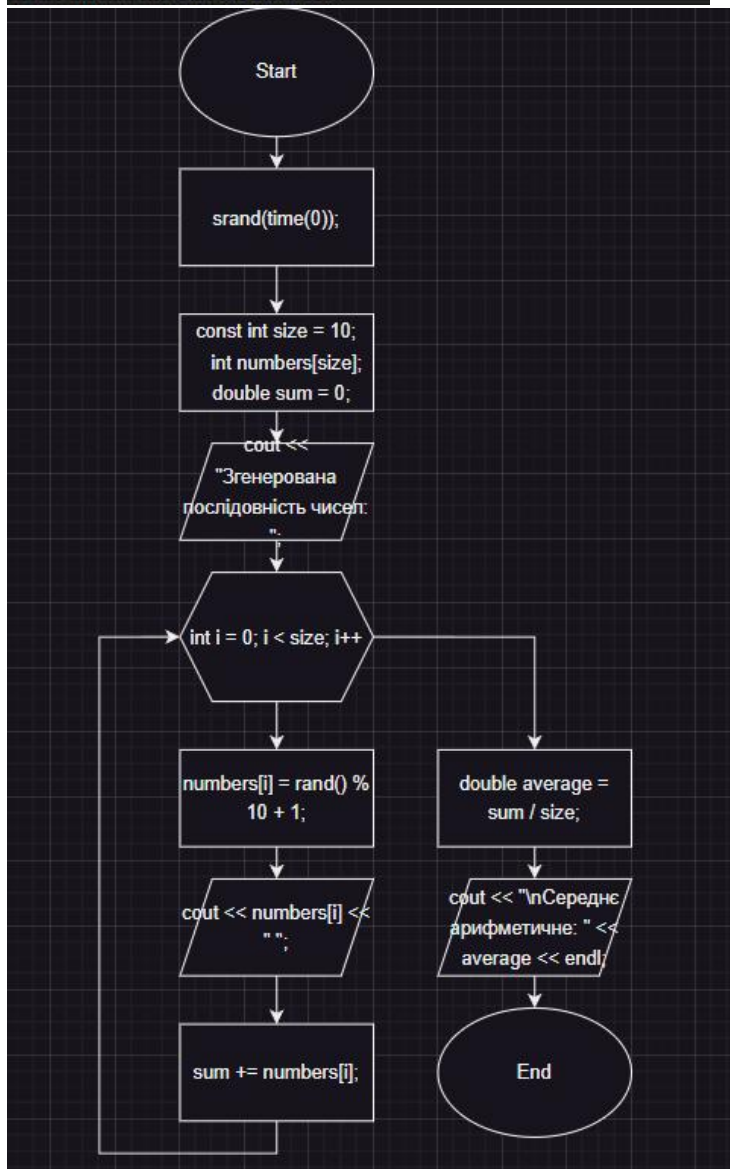
Завдання 4: VNS Practice Work Task 4

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.

```
1  #include <iostream>
2  #include <cstdlib>
3  #include <ctime>
4  using namespace std;
5
6  int main() {
7      srand(time(0));
8
9      const int size = 10;
10     int numbers[size];
11     double sum = 0;
12
13     cout << "Згенерована послідовність чисел: ";
14
15     for (int i = 0; i < size; i++) {
16         numbers[i] = rand() % 10 + 1;
17         cout << numbers[i] << " ";
18         sum += numbers[i];
19     }
20
21     double average = sum / size;
22
23     cout << "\nСереднє арифметичне: " << average << endl;
24
25     return 0;
26 }
27
```

Згенерована послідовність чисел: 6 8 4 1 7 7 1 5 3 7
Середнє арифметичне: 4.9
PS C:\Users\supes\projects>

Згенерована послідовність чисел: 10 7 10 1 7 7 4 9 2 6
Середнє арифметичне: 6.3
PS C:\Users\suepec\projects>



Завдання 5: Algotester Practice Task 1

Перехід дороги

Обмеження: 2 сек., 256 МБ

Зенік і Марічка наступного року складають ЗНО. Готуй сани влітку, а воза — взимку. Тому вони хочуть почати підготовку якнайшвидше. Аби підготовка була цікавішою, вони вирішили запропонувати своїм однокласникам приєднатися. Їхні однокласники не поспішають готуватися до ЗНО, а натомість готуються до олімпіад з інформатики на Алготестері.

Зенік і Марічка мають відвідати кожного однокласника і переконати в тому, що треба готуватися до ЗНО, добре його скласти, вступити до університету, а там уже можна готуватися і до студентських олімпіад. На щастя, їм не доведеться ходити по всьому місту, адже всі однокласники живуть на одній вулиці. З одного боку вулиці всі будинки мають парні номери, а з іншого — непарні.

Усього треба відвідати n однокласників. Зенік і Марічка будуть відвідувати їх у такому порядку, в якому вони записані у списку учнів класу. i -ий у списку однокласник мешкає в будинку з номером a_i .

Зенік і Марічка пропонують вам теж приєднатися до підготовки. Для початку розв'яжіть простеньку задачку — порахуйте, скільки разів їм доведеться перейти дорогу.

Вхідні дані

У першому рядку задано одне ціле число n — кількість однокласників, яких треба відвідати.

У другому рядку задано n цілих чисел a_i — номери будинків, у яких мешкають однокласники.

Вихідні дані

У єдиному рядку виведіть одне ціле число — відповідь на задачу.

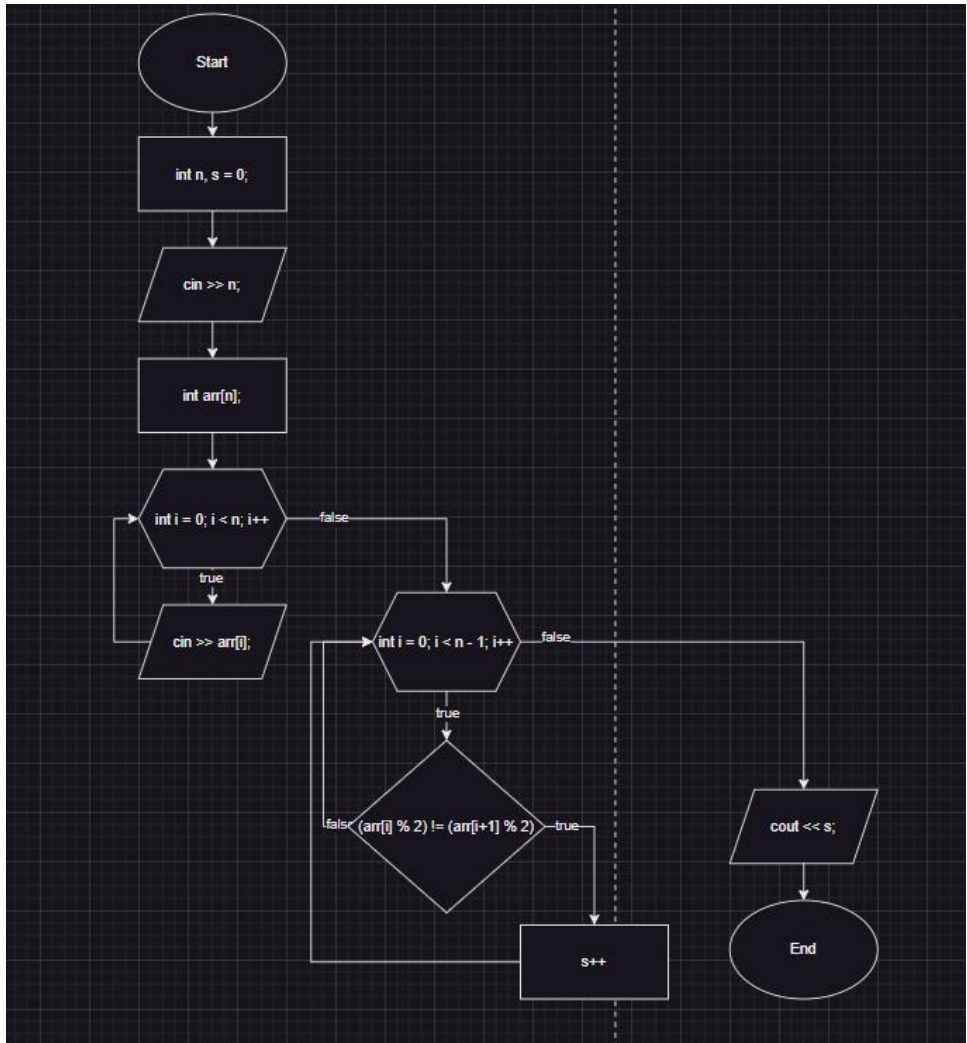
Обмеження

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

```
1 #include <bits/stdc++.h>
2 using namespace std;
3
4 //Перехід дороги
5 int main() {
6     int n, s = 0;
7     cin >> n;
8     int arr[n];
9     for (int i = 0; i < n; i++){
10         cin >> arr[i];
11     }
12     for (int i = 0; i < n - 1; i++){
13         if ((arr[i] % 2) != (arr[i+1] % 2)){
14             s++;
15         }
16     }
17     cout << s;
18     return 0;
19 }
```

```
7
4 44 7 47 74 47 74
4
PS C:\Users\supec\projects> |
```

годину тому	1401 - Перехід дороги	C++ 14	Зараховано	0.050	2.391	1908224
-------------	-----------------------	--------	------------	-------	-------	---------



Завдання 6: Algotester Practice Task 2

Допоможе чи заб'є?

Обмеження: 2 сек., 256 МБ

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.

Обмеження

$1 \leq |s|, k \leq 10^5$.

Приклади

```

1  #include <bits/stdc++.h>
2  #include <string>
3  #include<iostream>
4  using namespace std;
5
6  //Допоможе
7
8  int main() {
9      int n, k;
10     cin >> n;
11     k = 0;
12     string a;
13     cin.ignore();
14     getline(cin, a);
15     for(int i = 0; i < a.length();){
16         if (a[i] == 'T' && a[i+1] == 'O' && a[i+2] == 'I' && a[i+3] == 'L' && a[i+4] == 'E' && a[i+5] == 'T'){
17             i += 6;
18             k++;
19         }
20         else{
21             i++;
22         }
23     }
24     if (k >= n) {
25         cout << "YES";
26     }
27     else{
28         cout << "NO";
29     }
30
31     return 0;
32 }

```

```

2
HELPTOILETMENPLEASETOILET
YES

```

PS C:\Users\suepc\projects>

```

3
HELPTOILETMENPLEASETOILET
NO

```

PS C:\Users\suepc\projects>

годину тому

0081 - Допоможе чи заб'є?

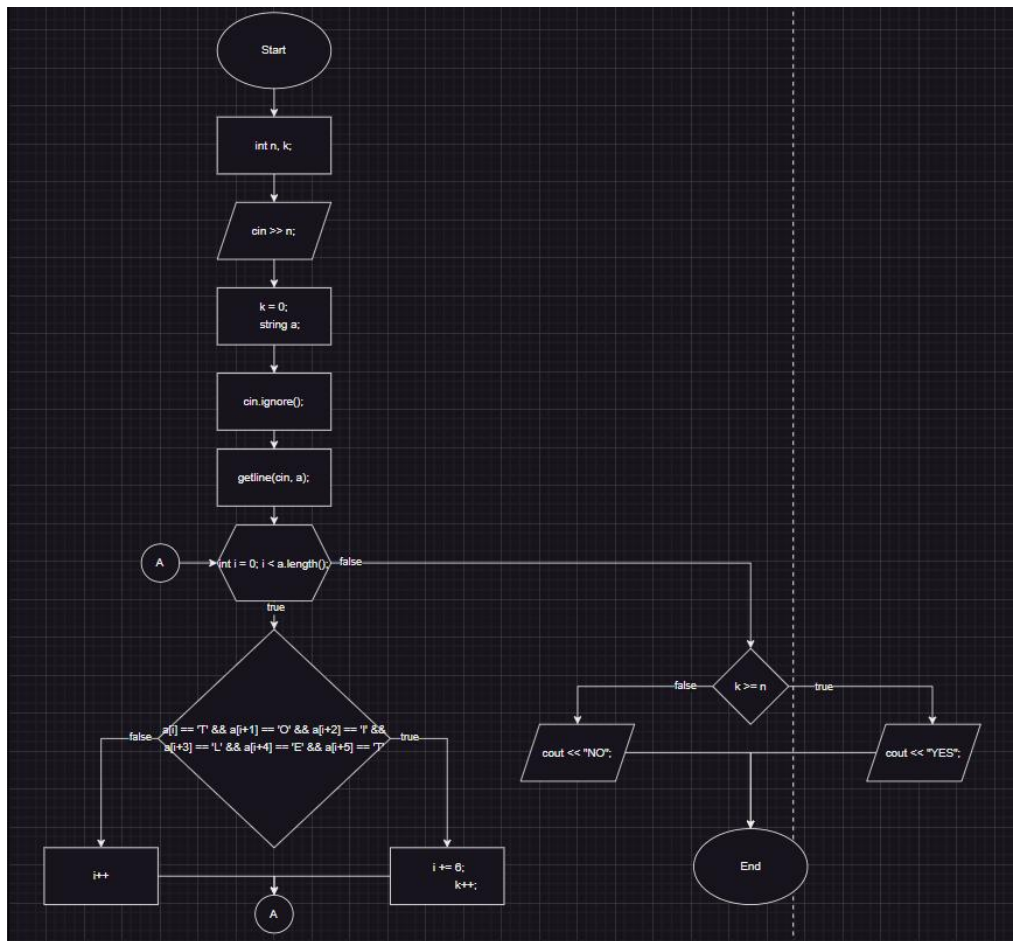
C++ 14

Зараховано

0.006

1.191

1908223



Завдання 7: Algotester Practice Task 3

Зуби

Обмеження: 2 сек., 256 МБ

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 Бісеняті є n зубів. Кожен зуб має коефіцієнт загостреності a_i . Також існує межа загостреності k . 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 n та k — кількість зубів та межа загостреності відповідно.

В другому рядку задано n цілих чисел a_i — коефіцієнти загостреності зубів.

Вихідні дані

Єдине ціле число — відповідь на задачу.

Обмеження

$$1 \leq n \leq 10^5,$$

$$1 \leq k \leq 10^9,$$

$$1 \leq a_i \leq 10^9.$$

```

1  #include <iostream>
2  using namespace std;
3
4  //Зуби
5
6  int main() {
7      int n, k, maxi = 0, now = 0;
8      cin >> n >> k;
9
10     for (int i = 0; i < n; i++) {
11         int ai;
12         cin >> ai;
13         if (ai >= k) {
14             now++;
15             maxi = max(maxi, now);
16         } else {
17             now = 0;
18         }
19     }
20
21     cout << maxi << endl;
22
23     return 0;
24 }

```

```

7 4
7 1 4 7 6 3 4
3
PS C:\Users\supec\projects>

```

11 днів тому

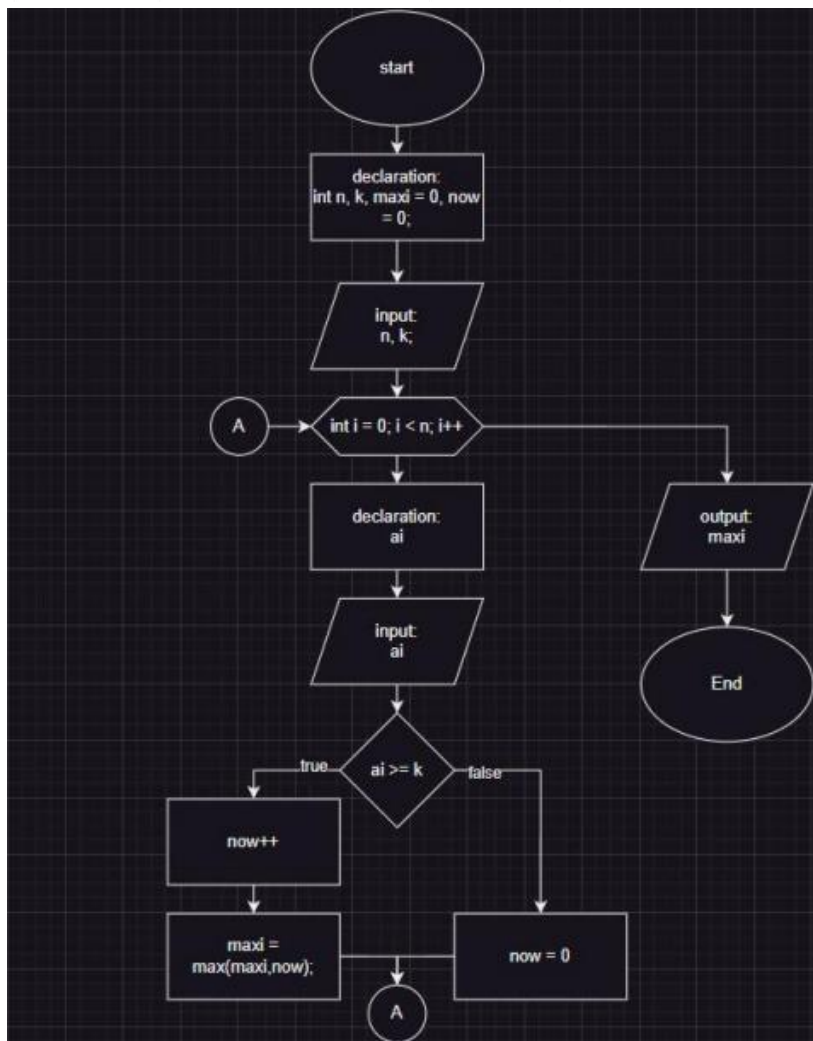
0182 - Зуби

C++ 14

Зараховано

0.049

1.102 1874655



Завдання 8: Algotester Practice Task 4

Зробити паліндром

Обмеження: 2 сек., 256 МБ

Задано рядок s . Дозволено переставляти літери рядка s між собою.

Чи можна перетворити рядок s на паліндром?

Паліндромом називається рядок, який однаково читається в обох напрямках (зліва направо та справа наліво).

Вхідні дані

У єдиному рядку задано s .

Вихідні дані

У єдиному рядку виведіть відповідь на задачу — слово **YES**, якщо утворити паліндром можна, або **NO**, якщо отримати паліндром не вийде.

Обмеження

Рядок s складається з малих літер латинського алфавіту.

40% тестів: $1 \leq |s| \leq 100$,

60% тестів: $100 \leq |s| \leq 10^6$.

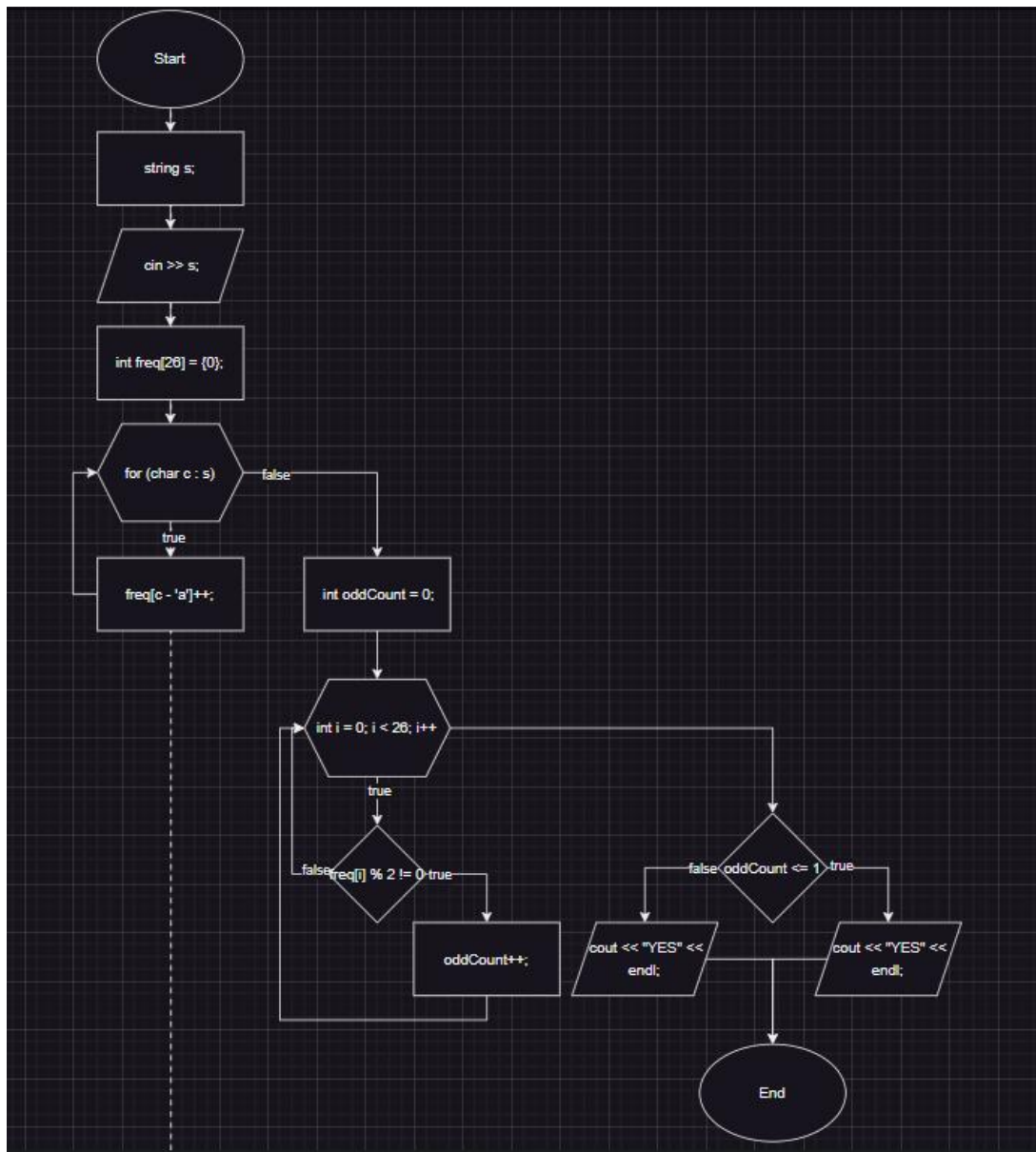
Приклади

Вхідні дані (stdin)	Вихідні дані (stdout)
aabbbb	YES

```
1  #include <iostream>
2  using namespace std;
3
4  //Зробити паліндром
5
6  int main() {
7      string s;
8      cin >> s;
9
10     int freq[26] = {0};
11
12     for (char c : s) {
13         freq[c - 'a']++;
14     }
15
16     int oddCount = 0;
17     for (int i = 0; i < 26; i++) {
18         if (freq[i] % 2 != 0) {
19             oddCount++;
20         }
21     }
22
23     if (oddCount <= 1) {
24         cout << "YES" << endl;
25     } else {
26         cout << "NO" << endl;
27     }
28
29     return 0;
30 }
```

```
aabbbb
YES
PS C:\Users\suepc\projects>
```

годину тому	0331 - Зробити паліндром	C++ 14	Зараховано	0.006	1.066	1908227
-------------	--------------------------	--------	------------	-------	-------	---------



Pull request : https://github.com/artificial-intelligence-department/ai_programming_playground_2024/pull/494

Висновок: Під час виконання цієї розрахункової роботи, я одержав практичні навчки та закріпив знання, одержані протягом семестру.

