

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



Звіт

про виконання лабораторних та практичних робіт блоку № 2

На тему: « Лінійні алгоритми. Розгалужені алгоритми. Умовні та логічні оператори. Змінні. Константи. Типи даних. Розмір типів даних. Ввід вивід.

Базові операції та вбудовані функції.Коментарі.»

з дисципліни: «Основи програмування» до:

ВНС Лабораторної Роботи № 1

Алготестер Лабораторної Роботи № 1

Практичних Робіт до блоку № 2

Виконав:

Студент групи ІІІ-11

Мартин Максим Ігорович

Львів 2024

Тема роботи: Ознайомлення з різними системами числення, такими як двійкова, вісімкова, десяткова та шістнадцяткова. Ознайомлення з етапами компіляції: препроцесор, компіляція. Використання бібліотек (stdio, cmath, iostream) та їх роль у програмуванні. Основи роботи з функціями cin, cout, printf, scanf та форматуванням виводу. Важливість, види та правила написання ефективних коментарів в коді. Структура, особливості, умовні оператори (if, else, switch-case) та логічні оператори (AND, OR, NOT).

Мета роботи: Ознайомитися з етапами компіляції, роботою препроцесора та директивами, а також зрозуміти функції компілятора для оптимізації процесу написання програм. Засвоїти принципи оголошення змінних і констант, вибору відповідного типу даних (int, char, float, double). Навчитися використовувати стандартні бібліотеки C++ (iostream, cmath, stdio) для полегшення програмування та виконання стандартних операцій. Опанувати роботу з операторами вводу/виводу (cin, cout, printf, scanf), а також форматуванням виводу даних у різних типах. Розуміти важливість коментарів у програмуванні та навчитися ефективно коментувати код для полегшення його підтримки. Опанувати лінійні алгоритми, їхню структуру та написання послідовних операцій у програмах C++. Засвоїти використання розгалужень та умовних операторів для побудови складніших алгоритмів. Вивчити логічні оператори та застосовувати їх у програмних умовах для управління логікою виконання алгоритмів.

Теоретичні відомості:

1. Теоретичні відомості з переліком важливих тем:
 - Тема №1: Використання систем числення в програмуванні.
 - Тема № 2: Функції компілятора і етапи компіляції.
 - Тема № 3: Змінні, Константи, Типи даних та їх розміри.
 - Тема № 4: Бібліотеки в C++.
 - Тема № 5: Ввід та Вивід даних. Робота з різними типами даних у вводі/виводі.
 - Тема № 6: Коментарі у Коді. Види коментарів у C++.
 - Тема № 7: Розгалужені алгоритми та Умовні Оператори.

- Тема № 8: Визначення та види логічних операторів.

2. Індивідуальний план опрацювання теорії:

- Тема №1: Використання систем числення в програмуванні. •

[Джерела 1](#)

[Джерела 2](#)

Що опрацьовано:

- + Позиційна і непозиційна системи числення.
- + Система числення, що використовується в комп'ютерах.
- + Вісімкова система числення в Linux (права доступу до файлів).
- Статус: Ознайомлений
- Початок опрацювання теми: 15.10.2024.
- Звершення опрацювання теми: 15.10.2024 (25хв.).

- Тема №2: Функції компілятора і етапи компіляції. •

[Джерела 1](#)

[Джерела 2](#)

- Що опрацьовано:
 - + Компілятор та інтерпретатор – різниця.
 - + Шість фаз компілятора це 1) лексичний аналіз 2) синтаксичний аналіз 3) семантичний аналіз 4) проміжний генератор коду 5) оптимізатор коду 6) код Generator.
- Статус: Ознайомлений
- Початок опрацювання теми: 15.10.2024.
- Звершення опрацювання теми: 15.10.2024 (30хв.).

- Тема №3: Змінні, Константи, Типи даних та їх розміри. •

[Джерела](#)

- Що опрацьовано:
 - + Чотири типи даних, які зазвичай використовуються
 - + Типи констант (перед вісімковою 0, перед шістнадцятковою 0X, символні в одинарних лапках, рядкові в подвійних лапках).
 - + Оголошення констант.
 - + Float – 6-9 значень точності(частіше 7), double – 15-18 цифр(частіше 16), long double: 15, 18 чи 33.

- Статус: Ознайомлений
- Початок опрацювання теми: 30.10.2024.
- Звершення опрацювання теми: 01.11.2024.

Тема №4: Бібліотеки в C++.

- [Джерела](#)

Що опрацьовано:

Стандартна бібліотека C — кожний заголовок стандартної бібліотеки C включено у стандартну бібліотеку C++ під іншою назвою: вилучено закінчення .h і додано літеру 'c' на початку.

Наприклад, Cmath та math.h, якщо написати math.h, то використати змінну pow вже не можна, адже існує така функція, з Cmath таких проблем немає. Усі функції з cmath знаходяться у просторі імен std. Тому, щоб використовувати функцію pow, потрібно вказати, що це std::pow. Тому, можна використовувати ім'я pow як змінну в іншому контексті без конфліктів, оскільки компілятор знає, що є різниця між глобальною змінною і функцією з простору імен std.

- Статус: Ознайомлений
- Початок опрацювання теми: 01.11.2024.
- Звершення опрацювання теми: 01.11.2024 (35хв.).

Тема №5: Ввід та Вивід даних. Робота з різними типами даних у вводі/виводі.

[Джерела 1](#)

[Джерела 2](#)

[Джерела 3](#)

[Джерела 4](#)

Що опрацьовано:

Класи istream(Оператор вилучення >> значень з потоку), ostream (Оператор вставки << значень в потік) , iostream.

Потік вводу та виводу.

Специфікатори типів даних (%c - одиничний символ, %s рядок, %d та %i десяткове ціле число, %p - адреса (або вказівник), %f - число з рухомою комою і тд).

Scanf та printf структура. `Scanf("%d %f", &a, &b), printf("a: %d, b: %.2f\n", a, b);`

Статус: Ознайомлений

Початок опрацювання теми: 01.10.2024.

Звершення опрацювання теми: 02.10.2024 (20хв).

Тема №6: Коментарі у Коді. Види коментарів у C++.

- [Джерела](#)

Що опрацьовано:

- ✦ Однорядкові коментарі починаються з двох скісних рисок (//).
- ✦ Багаторядкові коментарі починаються з /* і закінчуються */
- Статус: Ознайомлений
- Початок опрацювання теми: 20.10.2024.
- Звершення опрацювання теми: 20.10.2024 (5хв.).

Тема №7: Розгалужені алгоритми та Умовні Оператори.

[Джерела 1](#)

[Джерела 2](#)

[Джерела 3](#)

- Що опрацьовано:
 - ✦ Умовні розгалуження if/else.
 - ✦ Оператор switch, лейбли case.
 - ✦ switch і fall-through, **switch і оператор break**
 - ✦ Тарнарний оператор.
- Статус: Ознайомлений
- Початок опрацювання теми: 02.11.2024.
- Звершення опрацювання теми: 02.11.2024 (25хв.).

Тема №8: Визначення та види логічних операторів.

- [Джерела](#)

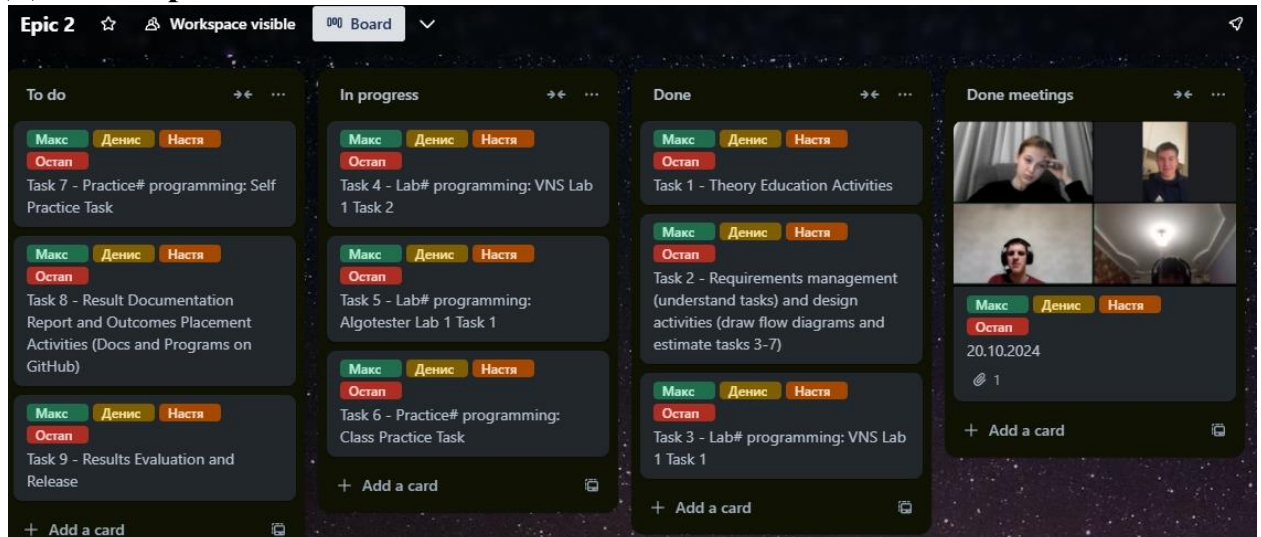
- Що опрацьовано:
 - ✦ Використання операторів І/АБО/НЕ
- Статус: Ознайомлений

Початок опрацювання теми: 03.11.2024.

Звершення опрацювання теми: 03.11.2024 (15хв.).

Виконання роботи:

Дошка Трело



Завдання №1 - VNS Lab 1 - Task 1- Variant 15 Задача:

Обчислити значення виразу при різних дійсних типах даних (float й double).

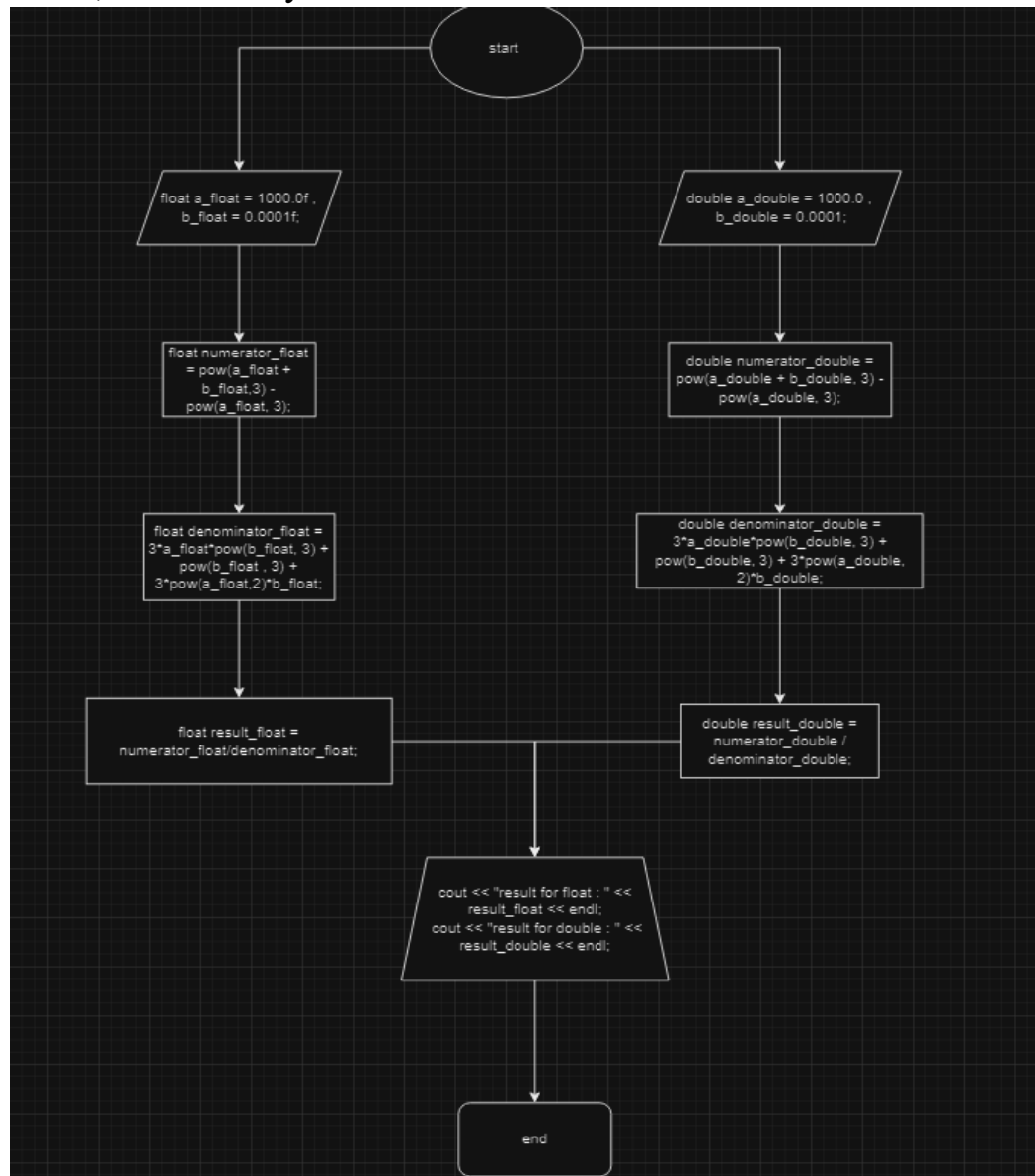
Обчислення варто виконувати з використанням проміжних змінних.

Порівняти й пояснити отримані результати.

$$\frac{(a+b)^3 - (a^3)}{3ab^2 + b^3 + 3a^2b},$$

при $a=1000$, $b=0.0001$

Блок схема , код та тестування:



```

1  #include <iostream>
2  #include <cmath>
3  using namespace std;
4  int main()
5  {
6      float a_float = 1000.0f , b_float = 0.0001f;
7      double a_double = 1000.0 , b_double = 0.0001;
8
9      //float
10     float numerator_float = pow(a_float + b_float, 3) - pow(a_float, 3);
11     float denominator_float = 3*a_float*pow(b_float, 3) + pow(b_float, 3) + 3*pow(a_float, 2)*b_float;
12     float result_float = numerator_float/denominator_float;
13
14     //double
15     double numerator_double = pow(a_double + b_double, 3) - pow(a_double, 3);
16     double denominator_double = 3*a_double*pow(b_double, 3) + pow(b_double, 3) + 3*pow(a_double, 2)*b_double;
17     double result_double = numerator_double / denominator_double;
18
19     cout << "result for float : " << result_float << endl;
20     cout << "result for double : " << result_double << endl;
21     return 0;
22 }
  
```

```

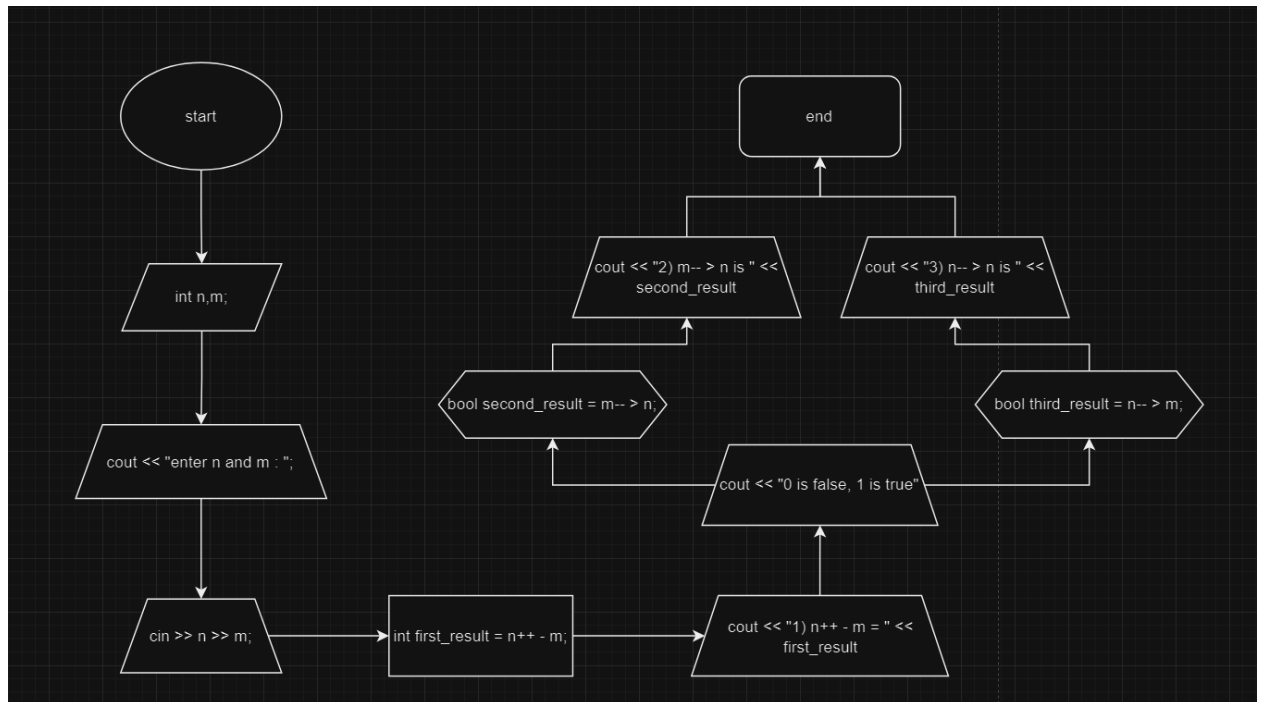
PS C:\Users\Maks\Documents\ai_programming_playground_2024\ai_11\maksym_martyn\epic_2> cd
_1 variant_15_maksym_martyn.cpp -o vns_lab_1_task_1_variant_15_maksym_martyn } ; if ($?)
result for float : 1.2207
result for double : 1
  
```

Завдання №2 - VNS Lab 1 - Task 2- Variant 15 Задача:

Обчислити значення виразів. Пояснити отримані результати

1. $m - ++n$
2. $++m > --n$
3. $--n < ++m$

Блок схема , код та тестування:



```
1  #include <iostream>
2  using namespace std;
3  int main ()
4  {
5      int n,m;
6      cout << "enter n and m : ";
7      cin >> n >> m;
8      int first_result = n++ - m;
9      cout << "1) n++ - m = " << first_result << endl; // n will be +1 after result cause n++ not ++n ;
10     cout << "0 is false, 1 is true" << endl;
11     bool second_result = m-- > n;
12     cout << "2) m-- > n is " << second_result << endl;
13     bool third_result = n-- > m;
14     cout << "3) n-- > n is " << third_result << endl;
15     return 0;
16 }
```

```
1) n++ - m = -1
0 is false, 1 is true
2) m-- > n is 0
3) n-- > n is 1
```


Завдання №3 – Algotester lab 1 Variant 2

Задача:

У вас є стіл, у якого є 4 ніжки, довжини яких вам дано. Ви хочете зробити ніжки рівної довжини, для цього ви відпиляєте d від кожної ніжки (тобто вам буде дано 4 числа, кожне з яких буде означати відпилювання від відповідної ніжки стола). Якщо під час відпилювання найдовша ніжка стола буде у 2 рази більша-рівна ніж найменша ніжка - стіл перевернеться, але відпилювати ніжки це вам не завадить. Тобто якщо $h_{\max} \geq 2 * h_{\min}$ то стіл перевертається. Увага, це може статися і між початком та кінцем відпилювання, наприклад коли відпиляють 2, але ще не встигнуть відпиляти 3тю ніжку. Також ми вважаємо що перед відпилюванням стіл не перевернеться. Ваше завдання сказати чи після усіх маніпуляцій стіл буде цілий та паралельний підлозі. Якщо довжина, яку відріжуть буде більша за довжину ножки - вам треба вивести ERROR. Увага! Навіть якщо стіл перевернеться - ви все одно відпилюєте ніжки і можете отримати ERROR.

Input

4 цілих числа $h_1, 2, 3, 4$ - довжини ніжок стола 4 цілих числа $d_1, 2, 3, 4$ - довжина, яку відпиляють від відповідної ножки

Output

YES - якщо стіл буде стояти паралельно площині підлоги та довжина найменшої ніжки не буде рівна нулю.

ERROR - у випадку якщо ви відпиляєте більшу довжину ніж має ножка

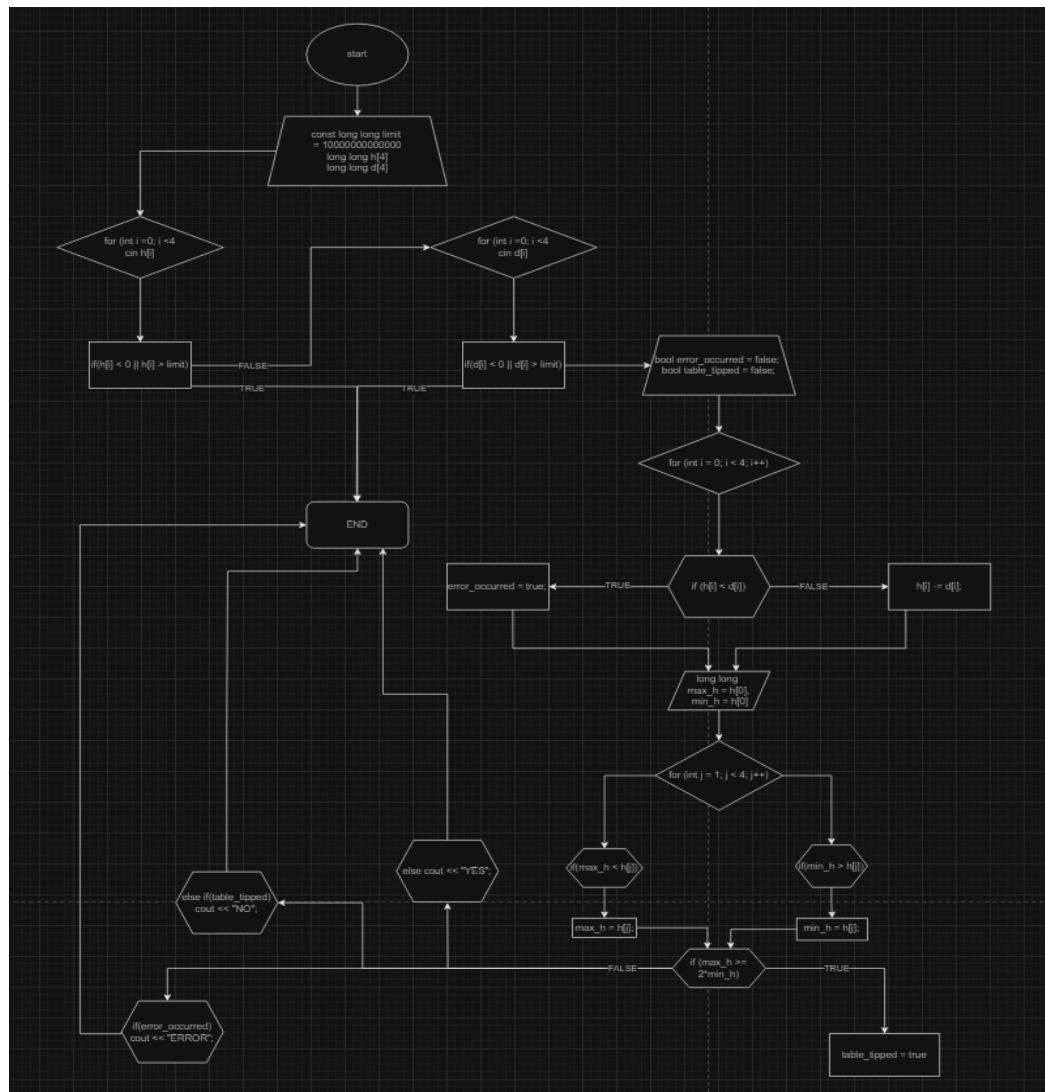
NO - у інших випадках

Constraints

$$0 \leq h_{1,2,3,4} \leq 10^{12}$$

$$0 \leq d_{1,2,3,4} \leq 10^{12}$$

Блок схема , код та тестування :



```

1  #include <iostream>
2  using namespace std;
3
4  int main ()
5  {
6      const long long limit = 10000000000000;
7      long long h[4];
8      long long d[4];
9
10     for (int i = 0; i < 4; i++)
11     {
12         cin >> h[i];
13         if(h[i] < 0 || h[i] > limit)
14             return 0;
15     }
16
17     for (int i = 0; i < 4; i++)
18     {
19         cin >> d[i];
20         if(d[i] < 0 || d[i] > limit)
21             return 0;
22     }
23
24     bool error_occurred = false;
25     bool table_tipped = false;

```

```

27     for (int i = 0; i < 4; i++)
28     {
29         if (h[i] < d[i])
30         {
31             error_occurred = true;
32         }
33         else
34         {
35             h[i] -= d[i];
36         }
37
38         long long max_h = h[0], min_h = h[0];
39
40         for (int j = 1; j < 4; j++)
41         {
42             if(max_h < h[j])
43                 max_h = h[j];
44
45             if(min_h > h[j])
46                 min_h = h[j];
47         }
48
49         if (max_h >= 2*min_h)
50         {
51             table_tipped = true;
52         }
53     }

```

```
55     if(error_occurred)
56         cout << "ERROR";
57
58     else if(table_tipped)
59         cout << "NO";
60
61     else
62         cout << "YES";
63
64     return 0;
65 }
66
67
```

```
_1_variant_2_maksym_martyn.cpp -o algotester_lab_1_variant_2_maksym_martyn } ; if ($?) { .\algotester_lab_1_variant_2_maksym_martyn }
10 10 10 10
5
6
4
1
NO
PS C:\Users\Maks\Documents\ai_programming_playground_2024\ai_11\maksym_martyn\epic_2>
```

Created	Compiler	Result	Time (sec.)	Memory (MiB)	Actions
10 minutes ago	C++ 23	Accepted	0.003	1.422	View

Завдання №4 - Class Practice Work - Особистий поради́ник

Задача Ви створюєте простий поради́ник щодо погоди. Користувач вводить поточні погодні умови, а програма видає рекомендації щодо активності на основі погоди.

Можливі варіанти погоди: • sunny; • rainy; • cloudy; • snowy; • windy;

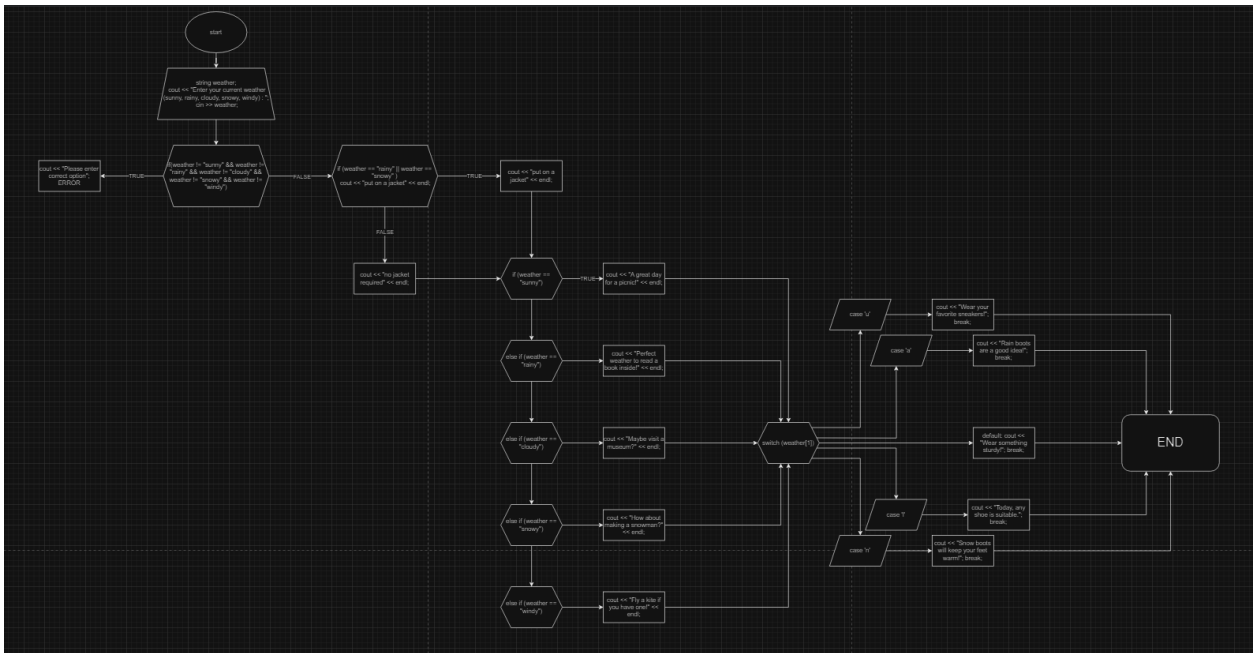
Мета Задачі Навчитися користуватися операторами галуження для структурування логіки програм.

Якщо користувач вводить будь-яку іншу умову, запропонуйте йому ввести дійсну умову.

Використовуйте таку логіку if else - щоб вирішити, чи повинен користувач взяти куртку чи ні. if, else if - щоб надати рекомендацію щодо активності (прогулянка, футбол, настільні ігри, etc).

switch case - для визначення типу рекомендованого взуття.

Блок схема , код та тестування:



```

1  #include <iostream>
2  using namespace std;
3  int main ()
4  {
5      string weather;
6      cout << "Enter your current weather (sunny, rainy, cloudy, snowy, windy) : ";
7      cin >> weather;
8      if(weather != "sunny" && weather != "rainy" && weather != "cloudy" && weather != "snowy" && weather != "windy")
9      {
10         cout << "Please enter correct option";
11         return 1;
12     }
13
14     if (weather == "rainy" || weather == "snowy" )
15     cout << "put on a jacket" << endl;
16
17     else
18     cout << "no jacket required" << endl;
19
20     if (weather == "sunny")
21     cout << "A great day for a picnic!" << endl;
22
23     else if (weather == "rainy")
24     cout << "Perfect weather to read a book inside!" << endl;
25
26     else if (weather == "cloudy")
27     cout << "Maybe visit a museum?" << endl;
28
29     else if (weather == "snowy")
30     cout << "How about making a snowman?" << endl;
31
32     else if (weather == "windy")
33     cout << "Fly a kite if you have one!" << endl;
  
```

```

35  switch (weather[1])
36  {
37      case 'u':
38          cout << "Wear your favorite sneakers!";
39          break;
40
41      case 'a':
42          cout << "Rain boots are a good idea!";
43          break;
44
45      case 'l':
46          cout << "Today, any shoe is suitable.";
47          break;
48
49      case 'n':
50          cout << "Snow boots will keep your feet warm!";
51          break;
52
53      default: cout << "Wear something sturdy!";
54      break;
55  }
56  return 0;
57  }

```

```

Enter your current weather (sunny, rainy, cloudy, snowy, windy) : rainy
put on a jacket
Perfect weather to read a book inside!
Rain boots are a good idea!

```

Завдання №5 Self Practice Task – Algotester Teeth

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 Бісеняти 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.

Input

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.

Output

Єдине ціле число — відповідь на задачу.

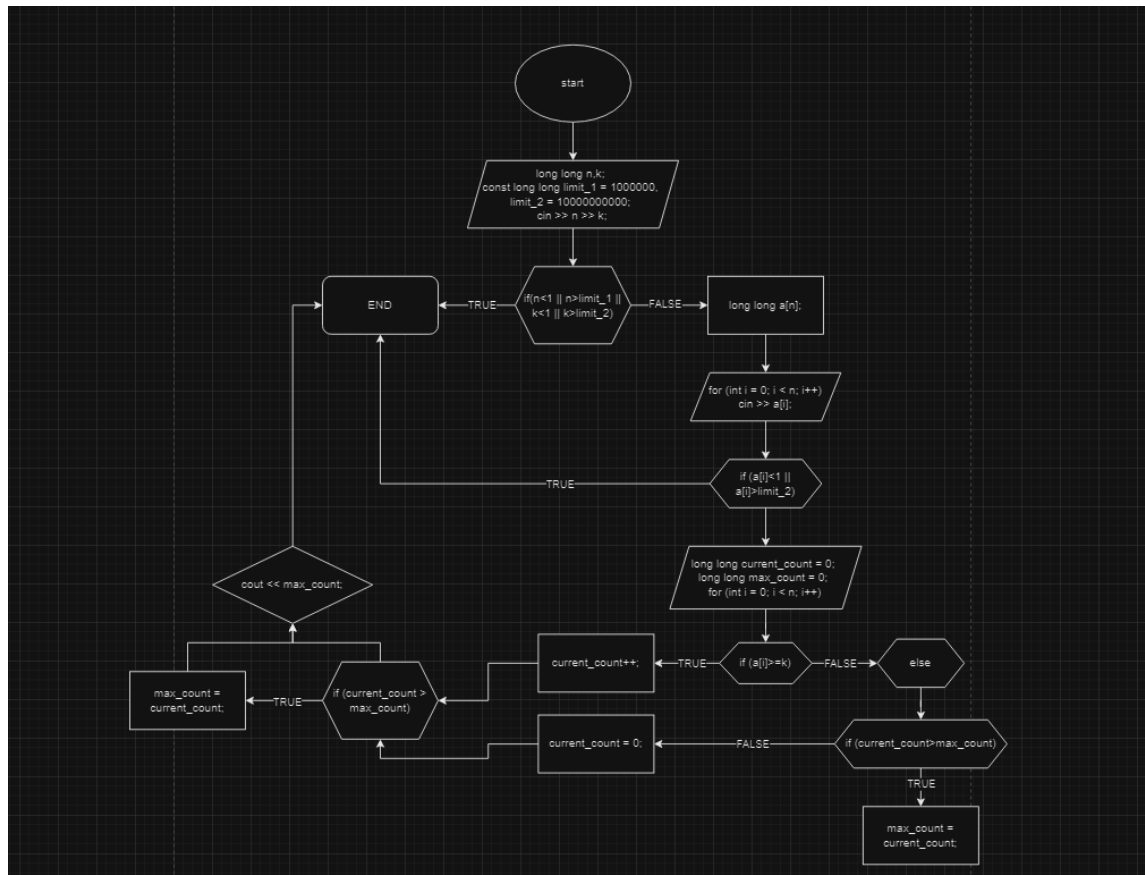
Constraints

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

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

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

Блок схема , код та тестування :



```

1 //https://algotester.com/uk/ArchiveProblem/DisplayWithEditor/20075
2 #include <iostream>
3 using namespace std;
4
5 int main ()
6 {
7     long long n,k;
8     const long long limit_1 = 1000000, limit_2 = 100000000000;
9     cin >> n >> k;
10    if(n<1 || n>limit_1 || k<1 || k>limit_2)
11    {
12        return 0;
13    }
14    cout << endl;
15
16    long long a[n];
17
18    for (int i = 0; i < n; i++)
19    {
20        cin >> a[i];
21        if (a[i]<1 || a[i]>limit_2)
22        {
23            return 0;
24        }
25    }
26    long long current_count = 0;
27    long long max_count = 0;

```

```

28     for (int i = 0; i < n; i++)
29     {
30         if (a[i]>=k)
31         {
32             current_count++;
33         }
34         else
35         {
36             if (current_count>max_count)
37             {
38                 max_count = current_count;
39             }
40             current_count = 0;
41         }
42     }
43
44     if (current_count > max_count)
45     {
46         max_count = current_count;
47     }
48
49     cout << max_count;
50     return 0;
51 }

```

```

PS C:\Users\Maks\Documents\ai_programming_playground_2024\ai_11\maksym_martyn\epic_2> cd "c:\User
work_algotester_task_1_maksym_martyn.cpp -o self_practice_work_algotester_task_1_maksym_martyn }
5 9
1 10 52 8 9
2

```

Створено	Компілятор	Результат	Час (сек.)	Пам'ять (МБ)	Дії
декілька секунд тому	C++ 23	Зарховано	0.044	2.008	Перегляд

Висновок: Під час виконання другого епіку я ознайомився з основними етапами компіляції та роботою препроцесора, що дало мені краще розуміння того, як працює компілятор та які функції він виконує для оптимізації коду. Я засвоїв принципи оголошення змінних і констант, а також вибір відповідних типів даних, що є важливими для ефективної роботи з пам'яттю. Зрозумів мету використання стандартних бібліотек, таких як `iostream` та `cmath`. Я оволодів навичками форматування виводу, написання лінійних алгоритмів і використання умовних операторів для побудови складніших програм

