

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

Виконала:

Студентка групи ІІІ-12
Лазаревич Юлія Дмитрівна

Львів 2024

Тема роботи:

Лінійні алгоритми. Розгалужені алгоритми. Умовні та логічні оператори. Змінні. Константи. Типи даних. Розмір типів даних. Ввід вивід. Базові операції та вбудовані функції. Коментарі.

Мета роботи:

Закріпити знання систем числення, ознайомитися з етапами та функціями компіляції, навчитися використовувати змінні, константи, визначати типи даних та їх розміри, навчитися використовувати базові бібліотеки мови C, опанувати базові операції та вбудовані функції, навчитися робити ввід та вивід даних та писати коментарі у коді, зрозуміти лінійні та розгалужені алгоритми, умовні та логічні оператори.

Теоретичні відомості:

1. Теоретичні відомості з переліком важливих тем:

Тема №1: Системи числення

Тема №2: Компіляція

Тема №3: Змінні, Константи, Типи даних та їх Розміри

Тема №4: Бібліотеки в C++

Тема №5: Ввід та Вивід даних

Тема №6: Базові Операції та Вбудовані Функції

Тема №7: Коментарі у Коді

Тема №8: Лінійні алгоритми

Тема №9: Розгалужені алгоритми та Умовні Оператори

Тема №10: Логічні Оператори

2. Індивідуальний план опрацювання теорії:

○ Тема №1: Системи числення.

○ Джерела Інформації:

- Лекції О. Пшеничного
- Практичні М. Фаріон
- Деякі [сайти про системи числення](#)

○ Що опрацьовано:

- Існує чотири типи:

- Десяткова (base 10) - використовується повсякденно. Цифри від 0 до 9. Кожна позиція в числі представляє степінь 10.
- Двійкова (base 2) - використовується в комп'ютерах. Складається з цифр 0 та 1. Кожна позиція в числі представляє степінь 2.
- Шістнадцяткова (base 16) - часто використовується у програмуванні. Складається з цифр від 0 до 9 та букв від A до F (де A = 10, B = 11, і т.д.). Кожна позиція в числі представляє степінь 16.
- Вісімкова (base 8) - використовується для більш зручного представлення двійкових чисел, оскільки три двійкових цифри можна представити однією вісімковою цифрою. Складається з цифр від 0 до 7. Кожна позиція в числі представляє степінь 8.
 - Як переводити з однієї системи числення в іншу
- Статус: ознайомлена.
- Початок опрацювання теми: 03.09.24
- Завершення опрацювання теми 17.09.24
- Тема №2: Компіляція.
 - Джерела Інформації:
 - Практичні М. Фаріон
 - Що опрацьовано:
 - Процес компіляції:
 - Preprocessing - виконується включення файлів та видалення коментарів.
 - Compilation - перетворення вихідного коду на мову асемблера.
 - Assembly - перетворення коду асемблера на машинний код та створення об'єктного файлу.
 - Linking - об'єднання об'єктних файлів і бібліотек у виконуваний файл.
 - Статус: ознайомлена.
 - Початок опрацювання теми: 9.09.24
 - Завершення опрацювання теми 20.09.24
- Тема №3: Змінні, Константи, Типи даних та їх Розміри
 - Джерела Інформації:
 - Лекції О. Пшеничного
 - Практичні М. Фаріон
 - Методичка з Лабі I
 - Що опрацьовано:
 - Їх визначення:

Змінні - іменовані області пам'яті, які мають свій тип даних та використовуються для зберігання даних цього типу.

Константи - змінні, які не можуть бути змінені після ініціалізації. Їх оголошення відбувається за допомогою ключового слова `const`.

Типи даних визначають, які значення можуть зберігатися в змінних. Вони також можуть бути беззнаковими (`unsigned`), що дозволяє зберігати лише невід'ємні числа.

- Основні типи даних:
 - `int`: цілі числа, 4 байти
 - `short`: цілі числа, 2 байти
 - `long`: цілі числа, 4 або 8 байтів
 - `long long`: цілі числа, 8 байтів
 - `float`: числа з плаваючою комою, 4 байти, менша точність
 - `double`: числа з плаваючою комою, 8 байтів, більша точність.
 - `long double`: числа з плаваючою комою, 8, 12 або 16 байтів, найвища точність.
- `char`: для символів/малих цілих чисел, 1 байт
- `bool`: для логічних значень(`true`/`false`), 1 байт
- Статус: ознайомлена.
- Початок опрацювання теми: 10.09.24
- Завершення опрацювання теми 20.09.24

○ Тема №4: Бібліотеки в C++

- Джерела Інформації:
 - Лекції О. Пшеничного
 - Практичні М. Фаріон
 - Окремі сайти та відео за темою
- Що опрацьовано:
 - Які існують бібліотеки(базові – `iostream`, `string`, `vector`, `algorithm`, `cmath`)
 - Приклад використання: `#include <iostream>`
- Статус: ознайомлена.
- Початок опрацювання теми: 17.09.24
- Завершення опрацювання теми 25.09.24

○ Тема №5: Ввід та Вивід даних

- Джерела Інформації:
 - Лекції О. Пшеничного
 - Практичні М. Фаріон

- Що опрацьовано:
 - Це основні операції. Приклад використання: `cin >> змінна; , cout << змінна << endl; .`
- Статус: ознайомлена.
- Початок опрацювання теми: 10.09.24
- Завершення опрацювання теми 17.10.24

- Тема №6: Базові Операції та Вбудовані Функції
 - Джерела Інформації:
 - Допомога чату джипіті
 - Сайт за темою
 - Що опрацьовано:
 - Базові операції: +, -, *, /, %
 - Логічні операції: **i (&&), або(||), не(!)**
 - Побітові операції: **&, |, <<, >>**
 - Вбудовані функції:
 - `sqrt(x)` - квадратний корінь з `x`.
 - `pow(x, y)`: x^y .
 - `max(a, b)`: більше значення з `a` та `b`.
 - `min(a, b)`: менше значення з `a` та `b`.
 - Статус: ознайомлена.
 - Початок опрацювання теми: 18.10.24
 - Завершення опрацювання теми 25.10.24

- Тема №7: Коментарі у Коді
 - Джерела Інформації:
 - Окремі сайти з туторіалом використання
 - Що опрацьовано:
 - Використовуються для пояснення функціоналу, можуть бути однорядковими (`//`) та багаторядковими (`/* */`).
 - Статус: ознайомлена.
 - Початок опрацювання теми: 02.09.24
 - Завершення опрацювання теми 02.10.24

- Тема №8: Лінійні алгоритми
 - Джерела Інформації:
 - Океме відео на ютубі
 - Пояснення чату джипіті
 - Що опрацьовано:
 - Лінійні алгоритми це база для багатьох програмних конструкцій. Вони виконуються послідовно, від початку до кінця, без відхилень або розгалужень.

- Статус: ознайомлена.
- Початок опрацювання теми: 30.09.24
- Завершення опрацювання теми 2.10.24

- Тема №9: Розгалужені алгоритми та Умовні Оператори

- Джерела Інформації:
 - Лекції О. Пшеничного
 - Практичні М. Фаріон
 - Окремі сайти за темою
- Що опрацьовано:
 - Розгалужені алгоритми використовують умовні оператори (if, else if, else, switch), щоб змінювати хід виконання програми залежно від певних умов.
- Статус: ознайомлена.
- Початок опрацювання теми: 20.09.24
- Завершення опрацювання теми 22.09.24

- Тема №10: Логічні Оператори

- Джерела Інформації:
 - Методичка до епіку
- Що опрацьовано:
 - Використовуються в умовних операторах і циклах для виконання логічних операцій, що дозволяють перевіряти і комбінувати булеві значення. Базові - логічне І (&&), логічне АБО (||), логічне НЕ (!)
- Статус: ознайомлена.
- Початок опрацювання теми: 30.09.24
- Завершення опрацювання теми 22.10.24

Виконання роботи:

1. Опрацювання завдання та вимог до програм та середовища:

Завдання №1 - VNS Lab 1 - Task 1- Variant 16.

Деталі завдання:

- Обчислити значення виразу при різних дійсних типах даних (float й double).
- Обчислення варто виконувати з використанням проміжних змінних.
- Порівняти й пояснити отримані результати.

$$\frac{(a - b)^3 - (a^3 - 3a^2b)}{b^3 - 3ab^2}, \text{ при } a = 1000, b = 0.0001$$

Завдання №2 - VNS Lab 1 - Task 2 - Variant 16.

Деталі завдання:

- Обчислити значення виразів. Пояснити отримані результати.

1) $++n * ++m$

2) $m++ < n$

3) $n++ > m$

Завдання №3 – Algotester Lab 1 - Variant 2.

Деталі завдання:

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.

Вхідні дані

- 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

Вихідні дані

- 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

Обмеження

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

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

Завдання №4 - Class Practice Work - Особистий поради́ник.

Деталі завдання:

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;

Вимоги:

Використати всі згадані в передумові задачі оператори галуження - if else, if, else if, switch case;

За потреби комбінувати оператори.

Завдання №5 - Self Practice Work –Щасливий результат(алготестер)

Деталі завдання:

Після завершення основного туру олімпіади з програмування Зеник отримав невеличкий клаптик паперу, на якому було надруковане число xx — кількість балів, що набрав Зеник. Зауважте, що згідно з кращими традиціями олімпіади з програмування, кількість балів Зеника не може бути нульовою чи від'ємною.

Помітивши не дуже щасливе обличчя Зеника, Марічка нагадала йому про щасливі цифри. Як ви вже напевно знаєте, щасливими вважають цифри 4 та 7. Марічка запевнила Зеника, що найкращим є не найбільший результат, а той, десятковий запис якого містить найбільше щасливих цифр.

Вам необхідно допомогти юному учаснику олімпіади з програмування та порахувати кількість щасливих цифр у його результаті.

Вхідні дані

У єдиному рядку задано одне ціле число xx — результат Зеника.

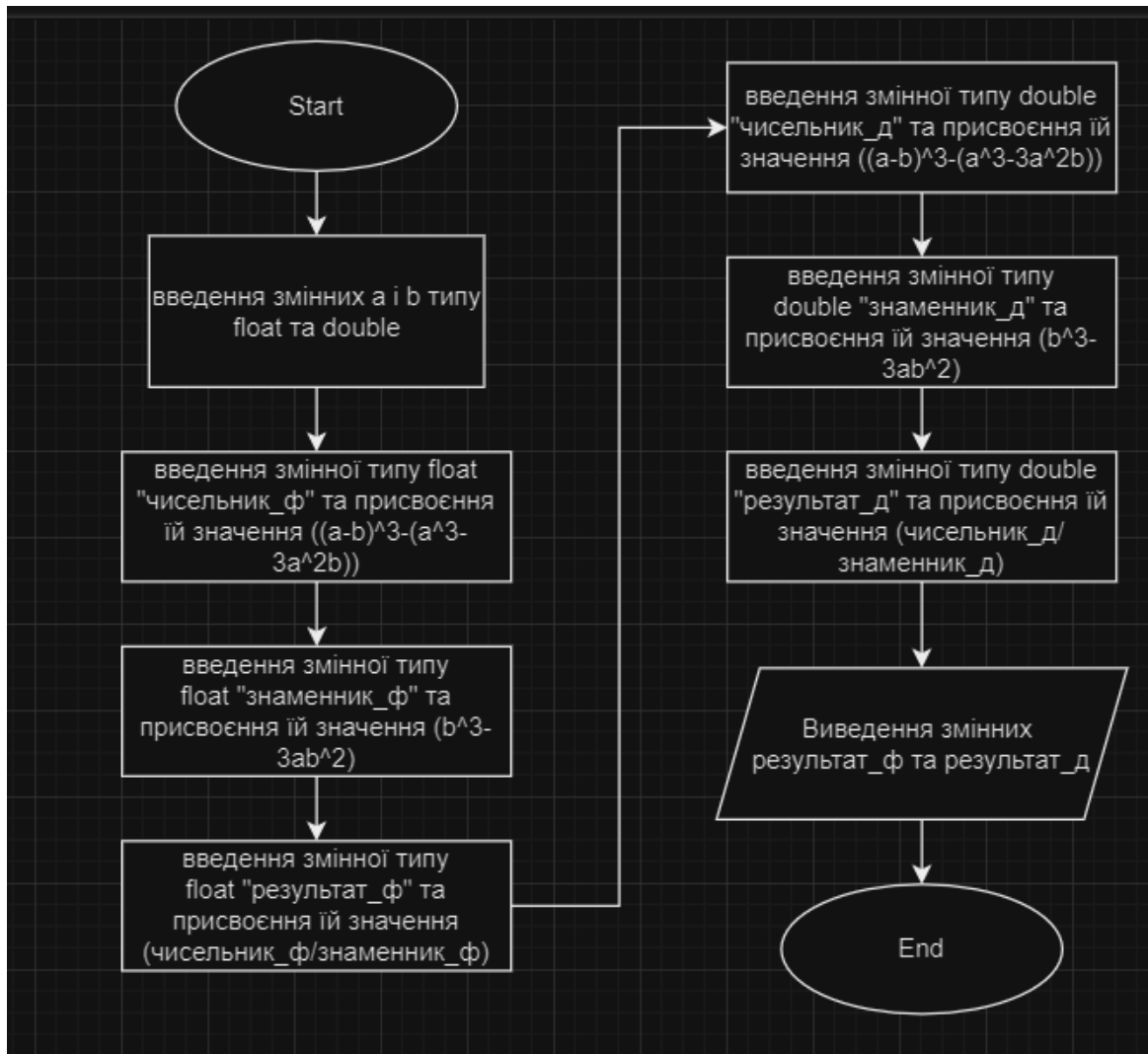
Вихідні дані

У єдиному рядку виведіть одне ціле число — кількість щасливих цифр у десятковому записі xx .

Обмеження $1 \leq x \leq 109$.

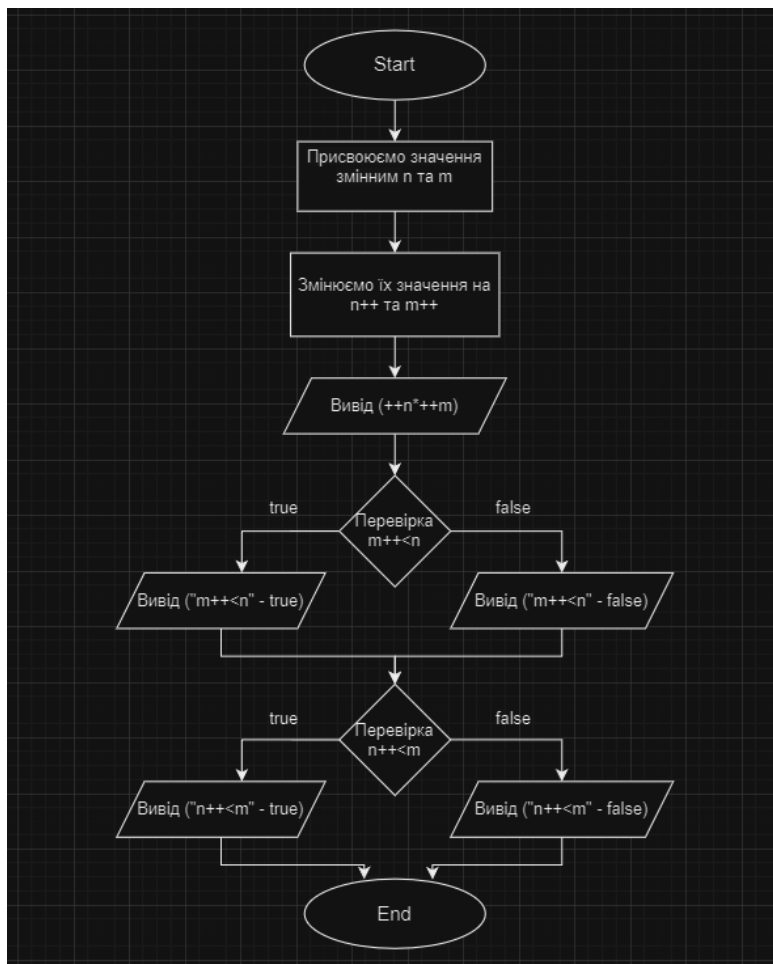
2. Дизайн та планована оцінка часу виконання завдань:

Програма №1 - VNS Lab 1 - Task 1- Variant 16.



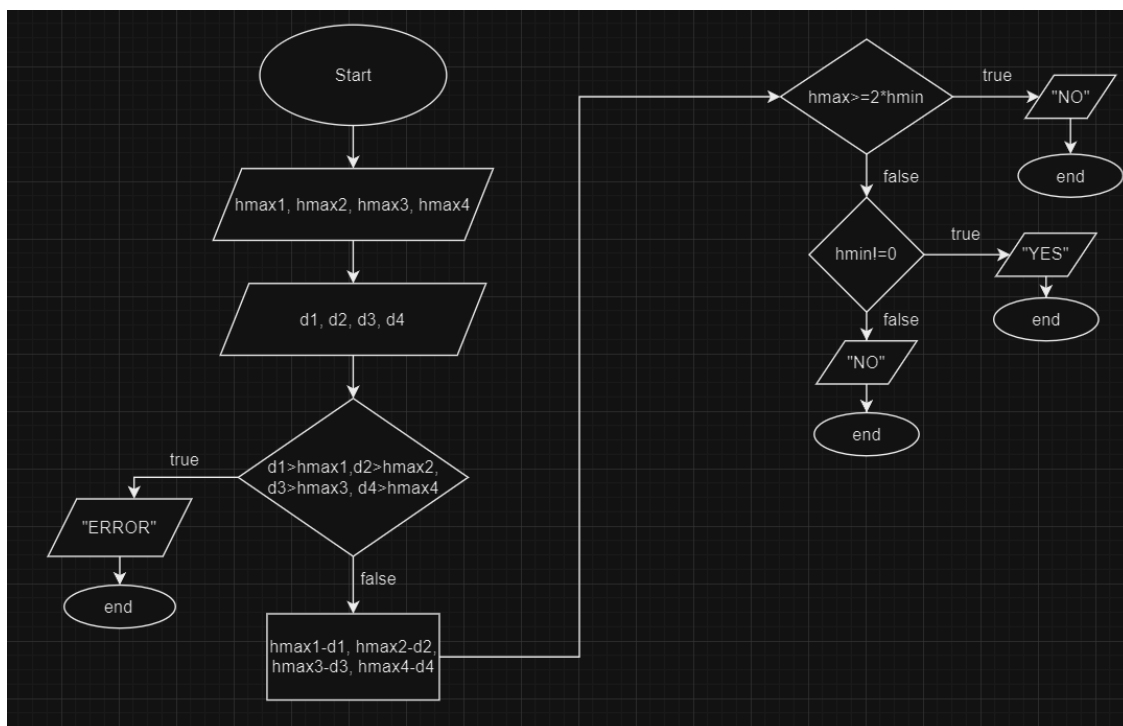
Плановий час на реалізацію: 30 хвилин

Програма №2 - VNS Lab 1 - Task 2- Variant 16.



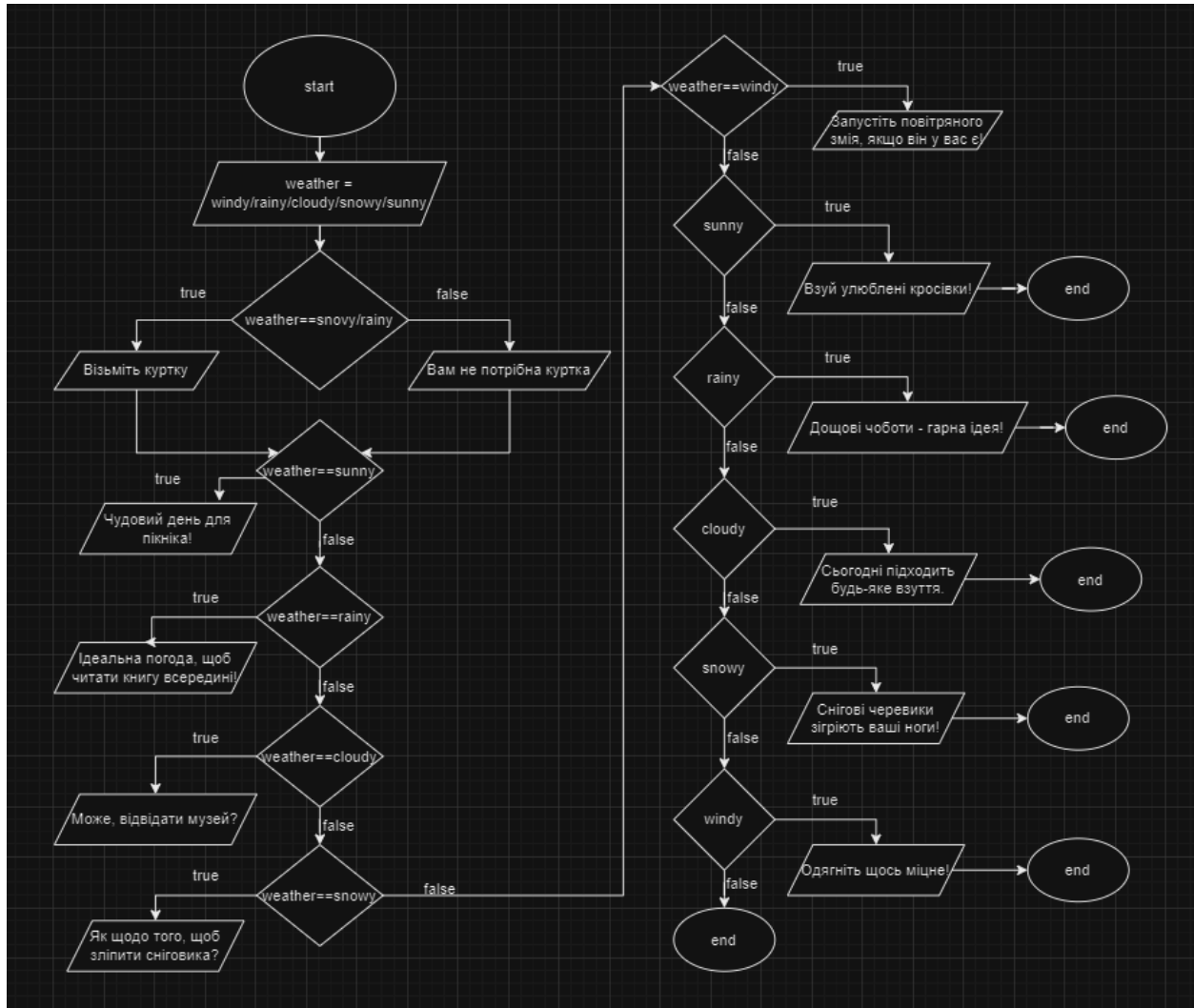
Плановий час на реалізацію: 30 хвилин

Програма №3 – Algotester Lab 1 - Variant 2.



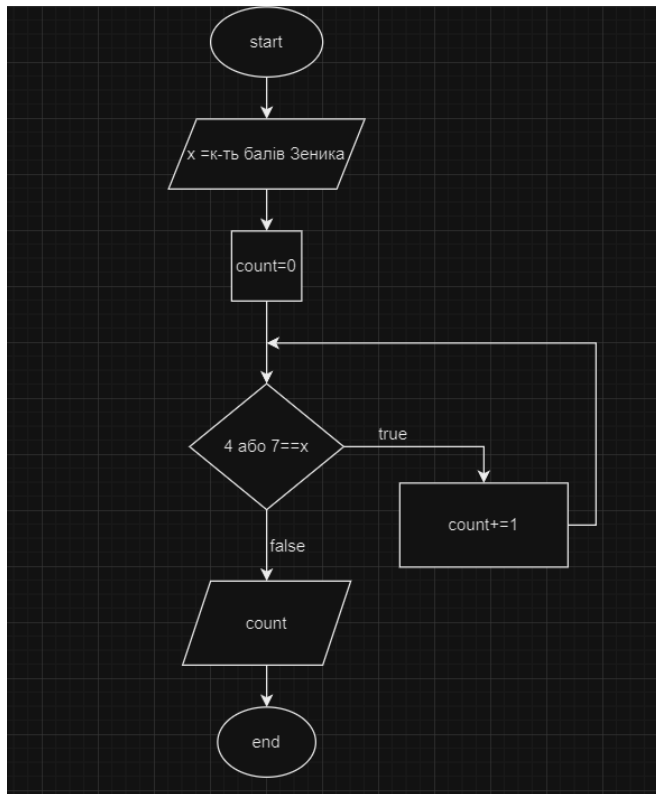
Плановий час на реалізацію: 1 година

Програма №4 - Class Practice Work - Особистий порадник.



Плановий час на реалізацію: 1 година

Програма №5 - Self Practice Work –Щасливий результат



Плановий час на реалізацію:
20 хвилин

3. Код програм з посиланням на зовнішні ресурси:

Завдання №1 - VNS Lab 1 - Task 1- Variant 16.

```

1  #include <iostream>
2  #include <cmath>
3  using namespace std;
4
5  int main() {
6      float a_f = 1000.0f, b_f = 0.0001f;
7      double a_d = 1000.0, b_d = 0.0001;
8
9      float numerator_f = pow(a_f - b_f, 3) - (pow(a_f, 3) - 3 * pow(a_f, 2) * b_f);
10     float denominator_f = pow(b_f, 3) - 3 * a_f * pow(b_f, 2);
11     float result_f = numerator_f / denominator_f;
12
13     double numerator_d = pow(a_d - b_d, 3) - (pow(a_d, 3) - 3 * pow(a_d, 2) * b_d);
14     double denominator_d = pow(b_d, 3) - 3 * a_d * pow(b_d, 2);
15     double result_d = numerator_d / denominator_d;
16
17     cout << "Результат типу float: " << result_f << "\nРезультат типу double: " << result_d << endl;
18
19     return 0;
20 }
  
```

ai_programming_playground_2024/ai_12/yuliia_lazarevych/epic_2/codes/vns_lab_1_task_1_variant_16_yuliia_lazarevych.cpp at epic_2_practice_and_labs_yuliia_lazarevych · artificial-intelligence-department/ai_programming_playground_2024

Завдання №2 - VNS Lab 1 - Task 2- Variant 16.

```

1  #include <iostream>
2  using namespace std;
3
4  int main() {
5      int n, m;
6      cout << "Введіть значення для n: ";
7      cin >> n;
8      cout << "Введіть значення для m: ";
9      cin >> m;
10
11     cout << "Результат виразу ++n*++m: " << ++n * ++m << endl;
12
13     bool result2 = m++ < n;
14     cout << "Результат виразу m++ < n: " << (result2 ? "true" : "false") << endl;
15
16     bool result3 = n++ > m;
17     cout << "Результат виразу n++ > m: " << (result3 ? "true" : "false") << endl;
18
19     return 0;
20 }

```

ai-programming-playground-2024.ai-12.yuliia-lazarevych/epic-2/codes/vns-lab-1-task-2-variant-16-yuliia-lazarevych.cpp at epic-2-practice-and-labs-yuliia-lazarevych · artificial-intelligence-department/ai-programming-playground-2024

Завдання №3 – Algotester Lab 1 - Variant 2.

```

1  #include <iostream>
2  #include <algorithm>
3
4  using namespace std;
5
6  int main() {
7      long long h[4], d[4];
8      cin >> h[0] >> h[1] >> h[2] >> h[3] >> d[0] >> d[1] >> d[2] >> d[3];
9
10     // перевіряємо, чи довжина відпилення була меншою за початкову довжину ніжки
11     for (int i = 0; i < 4; ++i) {
12         if (d[i] > h[i]) {
13             cout << "ERROR" << endl;
14             return 0;
15         }
16     }
17
18     // присвоюємо ніжкам оновлені значення, шукаємо мінімальну та максимальну ніжку і перевіряємо стіл на стійкість
19     for (int i = 0; i < 4; ++i) {
20         h[i] -= d[i];
21
22         long long hmin = *min_element(h, h + 4);
23         long long hmax = *max_element(h, h + 4);
24
25         if (hmax >= 2 * hmin || hmin == 0) {
26             cout << "NO" << endl;
27             return 0;
28         }
29     }
30
31     cout << "YES" << endl; // якщо в результаті зі столом все добре і він стоїть
32     return 0;
33 }

```

ai-programming-playground-2024.ai-12.yuliia-lazarevych/epic-2/codes/algotester-lab-1-variant-2-yuliia-lazarevych.cpp at epic-2-practice-and-labs-yuliia-lazarevych · artificial-intelligence-department/ai-programming-playground-2024

Завдання №4 - Class Practice Work - Особистий поради́ник.

```
1  #include <iostream>
2  #include <string>
3
4  using namespace std;
5
6  int main() {
7      string weather;
8      cout << "Введіть погоду (windy, rainy, cloudy, snowy, sunny): ";
9      cin >> weather;
10
11     cout << "Перевірка на потребу в куртці: ";
12     if (weather == "snowy" || weather == "rainy") {
13         cout << "Візьміть куртку" << endl;
14     } else {
15         cout << "Вам не потрібна куртка" << endl;
16     }
17
18     cout << "Рекомендація щодо активностей: ";
19     if (weather == "sunny") {
20         cout << "Чудовий день для пікніка!" << endl;
21     } else if (weather == "rainy") {
22         cout << "Ідеальна погода, щоб читати книгу всередині" << endl;
23     } else if (weather == "cloudy") {
24         cout << "Може, відвідати музей?" << endl;
25     } else if (weather == "snowy") {
26         cout << "Як щодо того, щоб зліпити сніговика?" << endl;
27     } else if (weather == "windy") {
28         cout << "Запустіть повітряного змія, якщо він у вас є" << endl;
29     } else {
30         cout << "Невідома погода" << endl;
31         return 0;
32     }
33
34     cout << "Рекомендація щодо взуття: ";
35     int weatherType = 0; //робимо змінні типу інт для свіч-кейсів
36     if (weather == "sunny") {
37         weatherType = 1;
38     } else if (weather == "rainy") {
39         weatherType = 2;
40     } else if (weather == "cloudy") {
41         weatherType = 3;
42     } else if (weather == "snowy") {
43         weatherType = 4;
44     } else if (weather == "windy") {
45         weatherType = 5;
```

```

46     }
47
48     switch (weatherType) {
49     case 1:
50         cout << "Взуй улюблені кросівки!" << endl;
51         break;
52     case 2:
53         cout << "Дощові чоботи - гарна ідея!" << endl;
54         break;
55     case 3:
56         cout << "Сьогодні підходить будь-яке взуття." << endl;
57         break;
58     case 4:
59         cout << "Снігові черевики зігріють ваші ноги!" << endl;
60         break;
61     case 5:
62         cout << "Одягніть щось міцніше!" << endl;
63         break;
64     default:
65         cout << "Немає рекомендацій щодо взуття" << endl;
66     }
67
68     return 0;
69 }
70

```

[ai-programming-playground 2024/ai 12/yuliia lazarevych/epic 2/codes/practice work task 1 yuliia lazarevych.cpp](https://ai-programming-playground-2024/ai-12/yuliia-lazarevych/epic-2/codes/practice-work-task-1-yuliia-lazarevych.cpp) at [epic 2 practice and labs yuliia lazarevych · artificial-intelligence-department/ai-programming-playground 2024](https://github.com/yuliia-lazarevych/epic-2-practice-and-labs-yuliia-lazarevych-artificial-intelligence-department-ai-programming-playground-2024)

Завдання №5 - Self Practice Work –Щасливий результат

```

1  #include <iostream>
2  #include <string>
3
4  using namespace std;
5
6  int main() {
7      string x;
8      cin >> x;
9      int count = 0;
10
11     for (char lucky : x) {
12         switch (lucky) {
13             case '4':
14             case '7':
15                 count++;
16                 break;
17         }
18     }
19
20     cout << count << endl;
21
22     return 0;
23 }

```

[ai-programming-playground 2024/ai 12/yuliia lazarevych/epic 2/codes/self practice work a lgotester task 1 yuliia lazarevych.cpp](https://ai-programming-playground-2024/ai-12/yuliia-lazarevych/epic-2/codes/self-practice-work-a-lgotester-task-1-yuliia-lazarevych.cpp) at [epic 2 practice and labs yuliia lazarevych · artificial-intelligence-department/ai-programming-playground 2024](https://github.com/yuliia-lazarevych/epic-2-practice-and-labs-yuliia-lazarevych-artificial-intelligence-department-ai-programming-playground-2024)

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

Завдання №1 - VNS Lab 1 - Task 1- Variant 16.

```
Результат типу float: 2.20703e+06  
Результат типу double: -1.00136
```

Витрачений час: 20 хвилин

Завдання №2 - VNS Lab 1 - Task 2- Variant 16.

```
Введіть значення для n: 14  
Введіть значення для m: 20  
Результат виразу ++n*++m: 315  
Результат виразу m++ < n: false  
Результат виразу n++ > m: false
```

Витрачений час: 30 хвилин

Завдання №3 – Algotester Lab 1 - Variant 2.

4	5	0
5	5	10
3	5	2
2	5	3
1	1	1
2	1	2
1	1	3
1	1	4
NO	YES	ERROR

Витрачений час: 1 година

Завдання №4 - Class Practice Work - Особистий порадишник.

```
Введіть погоду (windy, rainy, cloudy, snowy, sunny): windy  
Перевірка на потребу в куртці: Вам не потрібна куртка  
Рекомендація щодо активностей: Запустіть повітряного змія, якщо він у вас є  
Рекомендація щодо взуття: Одягніть щось міцніше!
```

Витрачений час: 1 година

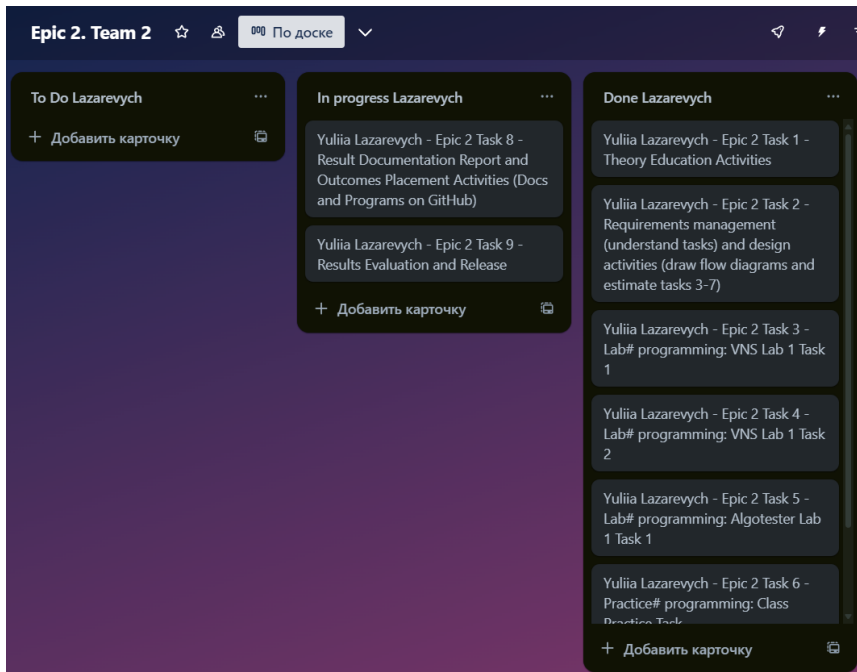
Завдання №5 - Self Practice Work –Щасливий результат

```
123456789  
2
```

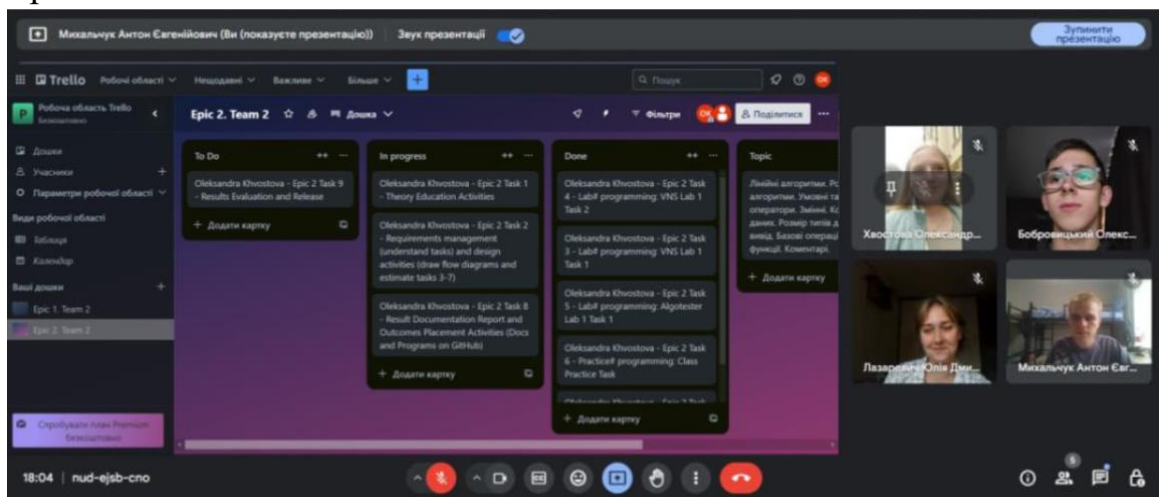
Витрачений час: 30 хвилин

6. Кооперація з командою:

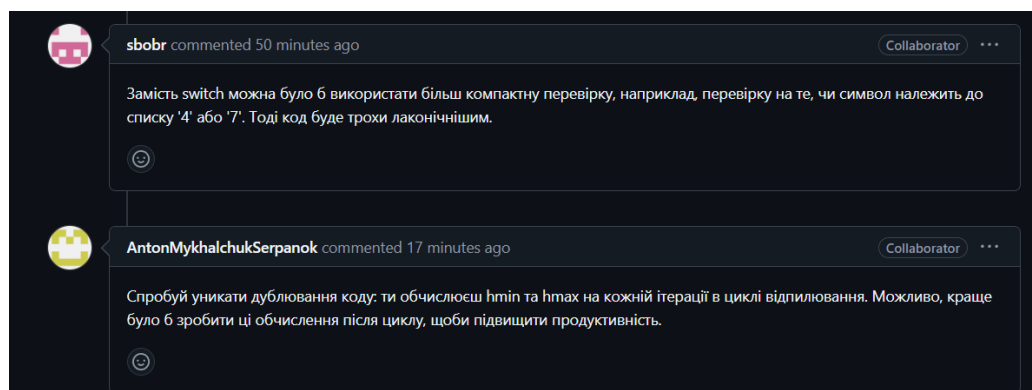
- Скрін прогресу по Трелло



- Скрін з 2-ї зустрічі по обговоренню задач Епіку та Скрін прогресу по Трелло



- Скрін з 2-ма коментарями від учасників команди на пул реквестів з Ревю Роботи



Висновки:

Виконуючи цей епік я закріпити знання систем числення, ознайомилася з етапами та функціями компіляції, навчилася використовувати змінні, константи, базові бібліотеки мови C , визначати типи даних та їх розміри, опанувала базові операції та вбудовані функції, навчилася робити ввід та вивід даних та писати коментарі у коді, зрозуміла лінійні та розгалужені алгоритми, умовні та логічні оператори.