

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



## Звіт

**про виконання лабораторних та практичних робіт блоку № 1**

На тему: «Програмування: алгоритм, програма, код. Системи числення.  
Двійкова система числення. Розробка та середовище розробки  
програми.»

**з дисципліни:** «Основи програмування»

до:

Практичних Робіт до блоку № 1

**Виконав:**

Студент групи ШІ-12  
Михальчук Антон Євгенійович

## Тема роботи:

Опанування інструментів та середовищ програмування для розробки C++. Робота з системами числення.

## Мета роботи:

Засвоїти основні інструменти та методи роботи з середовищами розробки, такими як Visual Studio Code, GitHub, Git а також оволодіти базовими знаннями роботи з системами числення та алгоритмами на C++. Додатково - вивчити принципи роботи з різними інструментами для управління проєктами. Крім того засвоїти теоретичний матеріал пов'язаний з системами числення.

## Теоретичні відомості:

- 1) Теоретичні відомості з переліком важливих тем:
  - Тема №\*.1: Package Managers OS.
  - Тема №\*.2: Console Commands в Linux подібному терміналі.
  - Тема №\*.3: Visual Studio Code.
  - Тема №\*.4: Дебагер та Лінер для C++.
  - Тема №\*.5: FlowCharts та Draw.io.
  - Тема №\*.6: Trello
  - Тема №\*.7: Google Docs
  - Тема №\*.8: Git, GitHub
  - Тема №\*.9: Algotester
  - Тема №\*.10: Системи числення
  - Тема №\*.11: C++ Basics
- 2) Індивідуальний план опрацювання теорії:
  - Тема №\*.1: Package Managers OS.
    - o Джерела Інформації
      - Відео. <https://www.youtube.com/watch?v=KHWIz1GnFJU>
      - Стаття. <https://www.msys2.org/docs/what-is-msys2/>
    - o Що опрацьовано:
      - Встановив msys2 та обрано ucrt64 середовище
      - Встановив gcc, скомпілював та запустив першу c++ програму
    - o Статус: Ознайомлений
    - o Початок опрацювання теми: 21.09.2024
    - o Звершення опрацювання теми: 21.09.2024
  - Тема №\*.2: Console Commands в Linux подібному терміналі.
    - o Джерела Інформації
      - Стаття. <https://kinsta.com/blog/linux-commands/>
    - o Що опрацьовано:
      - Використовуючи Git Bash, опрацював основні команди Linux подібного терміналу: ls, pwd, cd, rm, mv, mkdir, touch та інші.
    - o Статус: Ознайомлений
    - o Початок опрацювання теми: 22.09.2024
    - o Звершення опрацювання теми: 23.09.2024

- Тема №\*.3: Visual Studio Code.
  - o Джерела Інформації
    - Відео. <https://www.youtube.com/watch?v=w6PnOImop9Y>
    - Стаття. <https://code.visualstudio.com/docs/cpp>
  - o Що опрацьовано:
    - Встановив пакет розширень C/C++ Extension Pack та CMake для Visual Studio Code.
    - Використовую функцію IntelliSense, яка суттєво підвищила ефективність роботи завдяки інтелектуальному автодоповненню коду та наданню контекстних підказок.
    - За допомогою Code Runner зміг швидко запускати створений код.
  - o Статус: Ознайомлений
  - o Початок опрацювання теми: 23.09.2024
  - o Звершення опрацювання теми: 24.09.2024
  
- Тема №\*.4: Дебагер та Лінер для C++.
  - o Джерела Інформації
    - Відео. <https://www.youtube.com/watch?v=kK-Xo25DIXc>
    - Стаття. <https://learn.microsoft.com/en-us/shows/pure-virtual-cpp-2022/clang-tidy-in-visual-studio-code>
  - o Що опрацьовано:
    - Провів відлагодження коду за допомогою вбудованого Debugger.
    - Використав breakpoints та додаткові команди для детального аналізу.
    - Виявив та виправив помилки синтаксису за допомогою вбудованого Linter clang-tidy, також в нагоді знадобилася функція quickfix.
  - o Статус: Ознайомлений
  - o Початок опрацювання теми: 24.09.2024
  - o Звершення опрацювання теми: 25.09.2024
  
- Тема №\*.5: FlowCharts та Draw.io.
  - o Джерела Інформації
    - Відео. <https://www.youtube.com/watch?v=bN6i6dsoZTs>
    - Стаття. <https://en.wikipedia.org/wiki/Flowchart>
  - o Що опрацьовано:
    - Розібрався з різними видами блок схем та типами блоків.
    - Ознайомився з функціоналом Draw.io
  - o Статус: Ознайомлений
  - o Початок опрацювання теми: 25.09.2024
  - o Звершення опрацювання теми: 25.09.2024
  
- Тема №\*.6: Trello
  - o Джерела Інформації
    - Відео. <https://www.youtube.com/watch?v=geRKHfzTxNY>
  - o Що опрацьовано:
    - Ознайомився з інтерфейсом та функціоналом Trello

- Додав завдання з Епіку до tasks list
  - Протягом виконання завдань додав їх до done list
  - В об'єднаній дошці завдань групи відслідковував прогрес моїх колег
  - Статус: Ознайомлений
  - Початок опрацювання теми: 25.09.2024
  - Звершення опрацювання теми: 25.09.2024
- Тема №\*.7: Google Docs
  - Джерела Інформації
    - Відео. <https://www.youtube.com/watch?v=X7ShcCxOmns>
  - Що опрацьовано:
    - Я ознайомився з основними функціональними можливостями Google Docs.
    - Протягом роботи з Google Docs я детально вивчив і опанував основні інструменти для створення та редагування документів.
    - Я застосував Google Docs для створення звіту
  - Статус: Ознайомлений
  - Початок опрацювання теми: 22.09.2024
  - Звершення опрацювання теми: 23.09.2024
- Тема №\*.8: Git, GitHub
  - Джерела Інформації
    - Відео. <https://www.youtube.com/watch?v=tRZGeaHPoaw>
  - Що опрацьовано:
    - Встановив систему контролю версій Git та локальне середовище Git Bash
    - Вивчив основні команди та концепції (commit, branch, merge)
    - Створив обліковий запис на платформі GitHub та ініціалізував новий репозиторій
    - Здійснив успішну синхронізацію локального проекту з віддаленим репозиторієм на GitHub
    - Створено git репозиторію
    - Було залито проєкт на GitHub
    - Ознайомившись з додатковими функціональними можливостями GitHub, зокрема системою керування задачами (issues) та механізмом огляду та злиття коду (pull requests), отримав уявлення про ефективну роботу в команді за допомогою Git.
    - Створено ssh ключ
  - Статус: Ознайомлений
  - Початок опрацювання теми: 22.09.2024
  - Звершення опрацювання теми: 22.09.2024
- Тема №\*.9: Algotester
  - Джерела Інформації
    - Стаття. <https://algotester.com/en/Home/Help>
  - Що опрацьовано:
    - Ознайомився з функціоналом та інтерфейсом Algotester.
    - Ознайомився правила роботи Algotester

- Вирішив кілька завдань з Algotester
  - Статус: Ознайомлений
  - Початок опрацювання теми: 06.09.2024
  - Звершення опрацювання теми: 25.09.2024
- Тема №\*.10: Системи числення
  - Джерела Інформації
    - Стаття.<https://www.geeksforgeeks.org/number-system-and-base-conversions/>
  - Що опрацьовано:
    - Провів дослідження різних систем числення.
    - Детально розглянув принципи представлення чисел у двійковій, вісімковій, шістнадцятковій та інших системах числення.
    - Особливу увагу приділив переведенню чисел з однієї системи числення в іншу, а також операціям над числами в різних системах числення.
  - Статус: Ознайомлений
  - Початок опрацювання теми: 15.09.2024
  - Звершення опрацювання теми: 25.09.2024
- Тема №\*.11: C++ Basics
  - Джерела Інформації
    - Відео. <https://www.youtube.com/watch?v=2UDMGCCRCjo>
    - Стаття. <https://www.w3schools.com/cpp/>
  - Що опрацьовано:
    - Вивчив синтаксис та семантику мови C++.
    - Особливу увагу приділяв таким темам, як оголошення змінних, типи даних (цілі, з плаваючою точкою, логічні, символи), оператори, умови та цикли.
  - Статус: Ознайомлений
  - Початок опрацювання теми: 15.09.2024
  - Звершення опрацювання теми: 26.09.2024

## Виконання роботи:

### 1. Опрацювання завдання та вимог до програм та середовища:

Завдання №1 Обчислення складних відсотків за депозитом

- Деталі завдання: Обчислити складні відсотки для депозиту, який був відкритий в банку на певний період часу під фіксовані відсотки з різними варіантами виплати відсотків.
- Важливі деталі для врахування в імплементації програми:
  - Використати функції `scanf` та `printf` для зчитування і форматування вводу/виводу;
  - В кінці програма має вивести повну інформацію про вкладені кошти, загальну суму інвестиції і суму самого заробітку.

Завдання №2 Системи числення

- Деталі завдання: Дано цілі натуральних  $n$  та  $k$ . Завдання - знайти найменшу основу системи числення  $x$ , у якій кожна цифра числа  $n$  буде дорівнювати  $k$ .
- Важливі деталі для врахування в імплементації програми:

- о У єдиному рядку задано два натуральних числа  $n$  та  $k$  ( $k < 10$ ).
- о Якщо не існує жодної системи числення, що задовільняє умові, вивести - 1.

Завдання №3 Завдання на калькуляції в двійковій системі

- Деталі завдання:

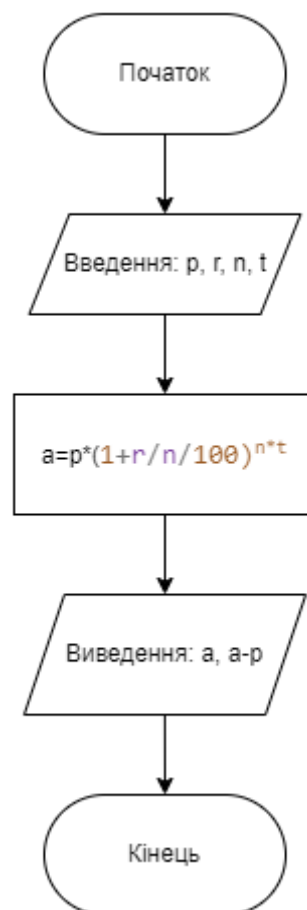
1	Згенерувати в рандомайзері десяткове число $u$ від 20 до 99
2	Згенерувати в рандомайзері десяткове число $x$ від 20 до 99
3	Перевести $u$ у двійкову систему числення
4	Перевести $x$ у двійкову систему числення
5	Додати два двійкових числа $x$ та $u$
6	Відняти від більшого двійкового числа менше двійкове число
7	Більше двійкове число поділити на менше двійкове число
8	Більше двійкове число помножити на менше двійкове число
9	Згенерувати в рандомайзері десяткове число $k$ від 20 до 99
10	Перевести $k$ у 16-ву систему числення

-

## 2. Дизайн та планована оцінка часу виконання завдань:

Програма №1 Обчислення складних відсотків за депозитом

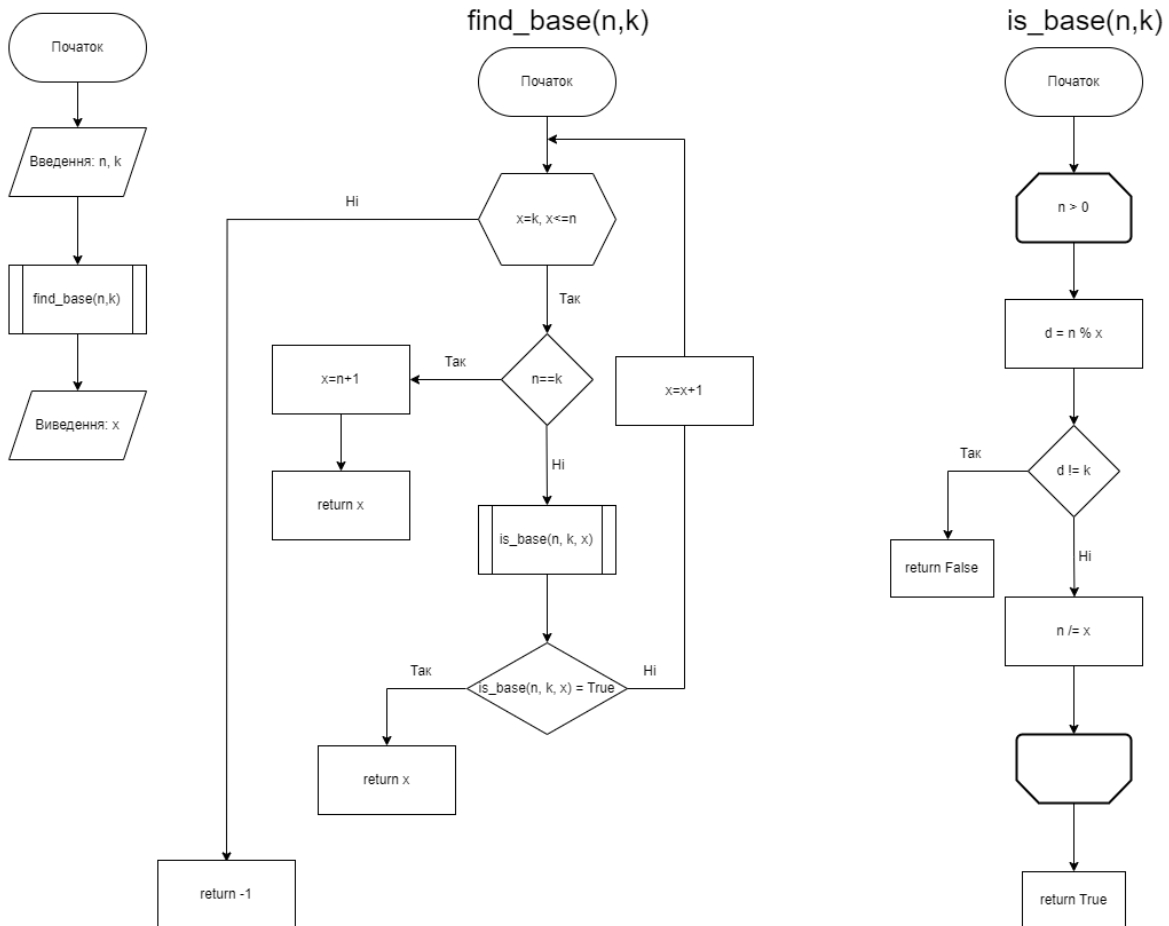
- Блок-схема :



- Планований час на реалізацію: 30 хв
- Важливі деталі для врахування в імплементації: використовувати scanf, printf та не забути перевести відсотки в десяткову систему.

Програма №2 Додаткове завдання: Системи числення

- Блок-схема



- Планований час на реалізацію: 1 г.
- Важливі деталі для врахування в імплементації:
  - o У єдиному рядку задано два натуральних числа n та k ( $k < 10$ ).
  - o Якщо не існує жодної системи числення, що задовільняє умові, вивести -1.

### 3. Конфігурація середовища до виконання завдань:

Завдання №1 Деталі по конфігурації середовища + скріншоти з підписами до скріншотів.

Під час налаштування робочого середовища було здійснено кілька ключових дій, спрямованих на забезпечення роботи з інструментами розробки програмного забезпечення.

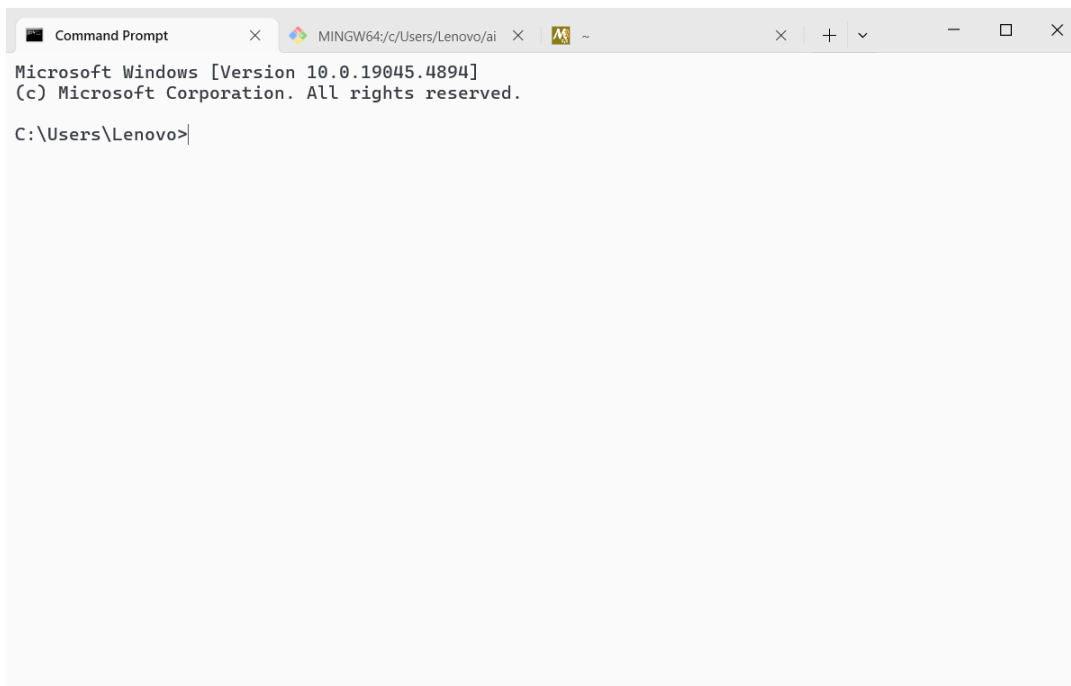
Спочатку було встановлено MSYS2, після чого обрано середовище ucrt64 для подальшої роботи. За допомогою цього середовища було встановлено компілятор GCC.

Для роботи з терміналом було налаштовано Git Bash, який дозволяє використовувати команди, подібні до тих, що використовуються у Linux.

Наступним кроком було встановлення Visual Studio Code та розширень для нього: C/C++ Extension Pack і CMake. Також було інтегровано Code Runner, що дозволяє запускати код безпосередньо з редактора.

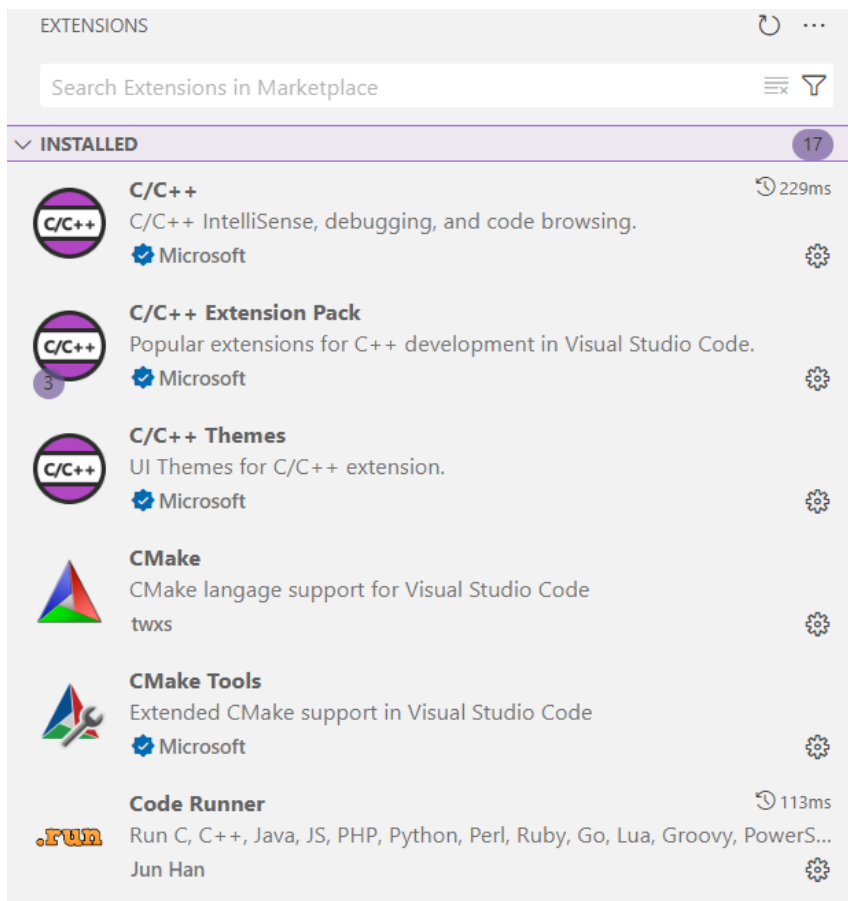
Окрім цього, було налаштовано вбудований Debugger у Visual Studio Code для відлагодження коду, а також Linter для автоматичної перевірки синтаксису і пошуку помилок.

Для управління версіями коду було встановлено Git і середовище Git Bash для взаємодії з Git. Також було створено обліковий запис на GitHub, ініціалізовано репозиторій, було створено гілку скопійованої репозиторії та було відправлено пул реквест на додавання моєї гілки з епіком до головного репозиторію.

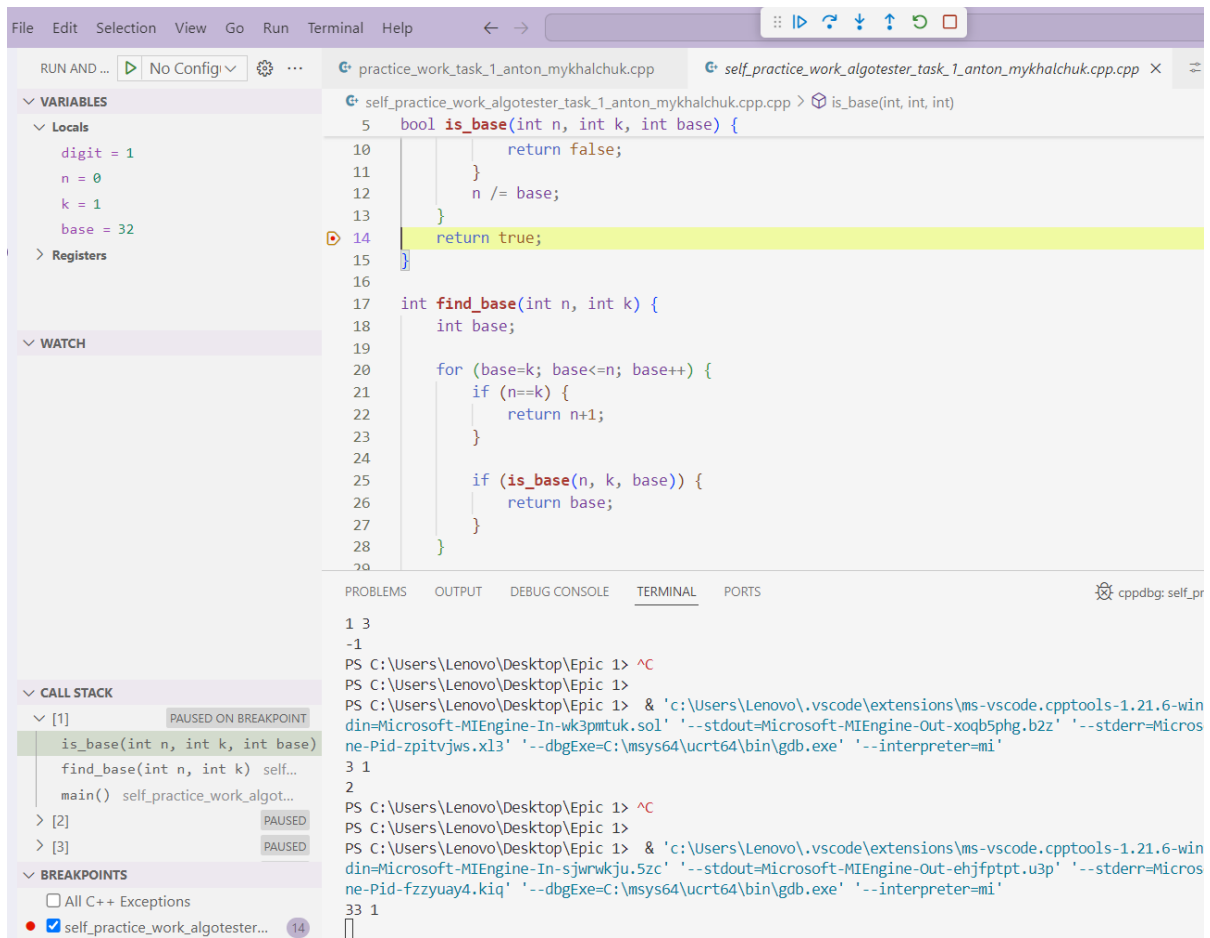


Встановленні середовища command prompt, git bash та msys2 у windows terminal

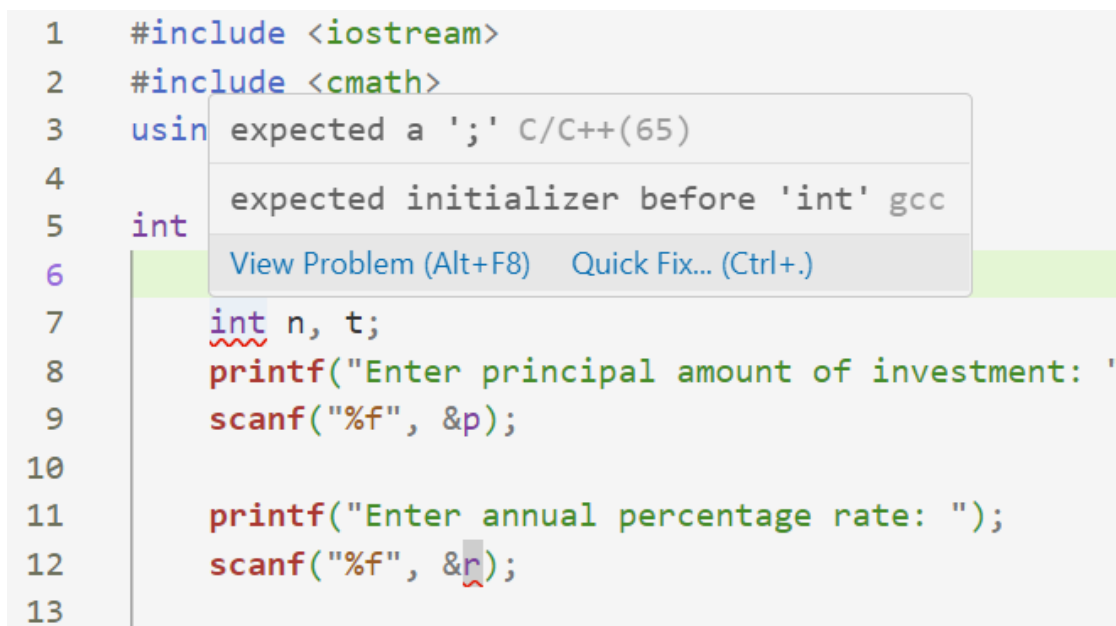




Встановленні розширення C/C++, C/C++ Extension Pack C/++ Themes, CMake, CMake Tools, Code Runner



Режим debugging з використанням breakpoints






Діагностика помилок синтаксису за допомогою Linter clang-tidy.

## SSH keys

[New SSH key](#)

This is a list of SSH keys associated with your account. Remove any keys that you do not recognize.

### Authentication keys

**Home Station**  
  
Added on Oct 2, 2024  
Last used within the last week — Read/write  


Check out our guide to [connecting to GitHub using SSH keys](#) or troubleshoot [common SSH problems](#).

Створено ssh key

## 4. Код програм з посиланням на зовнішні ресурси:

Завдання №1 Деталі по програмі + Вставка з кодом з підписами до вставки. Посилання на файл програми у пул-запиті GitHub

[https://github.com/artificial-intelligence-department/ai\\_programming\\_playground\\_2024/pull/14/commits/62fe4ab1eb381339e6521f327fd5cd30fb3b7f36#diff-46264ad872955422e4e34210db3f058f44ee114be3c319dc80407b01af437fef](https://github.com/artificial-intelligence-department/ai_programming_playground_2024/pull/14/commits/62fe4ab1eb381339e6521f327fd5cd30fb3b7f36#diff-46264ad872955422e4e34210db3f058f44ee114be3c319dc80407b01af437fef)

```
#include <iostream>    // Підключення бібліотеки для роботи з потоками
                        введення-виведення
#include <cmath>        // Підключення бібліотеки для математичних
                        операцій (для функції pow)

using namespace std;

int main() {
    // Оголошення змінних для збереження значень
    float a, p, r;      // a - загальна сума інвестицій, p - початкова
                        сума (principal), r - річна процентна ставка
    int n, t;           // n - кількість нарахувань процентів в рік, t -
                        кількість років інвестування

    // Введення початкової суми інвестицій від користувача
    printf("Enter principal amount of investment: ");
    scanf("%f", &p);    // Зчитування введеної користувачем початкової
                        суми (float)
```

```

// Введення річної процентної ставки від користувача
printf("Enter annual percentage rate: ");
scanf("%f", &r);    // Зчитування введеної процентної ставки (float)

// Введення кількості нарахувань процентів за рік
printf("Enter number of interest charges per year: ");
scanf("%d", &n);    // Зчитування кількості нарахувань процентів
(int)

// Введення часу інвестування в роках
printf("Enter time for which the money is invested, in years: ");
scanf("%d", &t);    // Зчитування кількості років інвестування (int)

// Обчислення загальної суми інвестицій за формулою складних
процентів
a = p * pow(1 + r / n / 100, n * t);
// pow(1 + r / n / 100, n * t) - підрахунок складних процентів
// r/n/100 - розрахунок процентної ставки на кожне нарахування
// n * t - загальна кількість нарахувань за весь час

// Виведення результату
printf("\n\n\nThe total amount of the investment, given:\nPrincipal
amount of the investment %.2f\nThe annual interest rate %.2f%\n\nThe
number of interest accruals per year %d\nand the time for which the
money is invested %d years\nWill be equal: %.2f\n", p, r, n, t, a);
// Виведення введених даних користувача і загальної суми інвестицій

// Виведення прибутку (сума інвестицій мінус початкова сума)
printf("The amount of the earnings themselves: %.2f", a - p);
// a - p - це чистий дохід від інвестицій

return 0;
}

```

Підпис та №1 до блоку з кодом програми

Завдання №2 Деталі по програмі + Вставка з кодом з підписами до вставки. Посилання на файл програми у пул-запиті GitHub

[https://github.com/artificial-intelligence-department/ai\\_programming\\_playground\\_2024/pull/14/commits/62fe4ab1eb381339e6521f327fd5cd30fb3b7f36#diff-b2798507a541c29cad04f26b69e4665c0cce6ee3e917dc6c7e01d5cca50182a0](https://github.com/artificial-intelligence-department/ai_programming_playground_2024/pull/14/commits/62fe4ab1eb381339e6521f327fd5cd30fb3b7f36#diff-b2798507a541c29cad04f26b69e4665c0cce6ee3e917dc6c7e01d5cca50182a0)

```
#include <iostream>    // Підключення бібліотеки для введення та
виведення даних

#include <string>        // Підключення бібліотеки для роботи зі строками
(не використовується в коді)

using namespace std;

// Функція перевіряє, чи можна представити число `n` в системі числення
з основою `base` так,

// щоб всі цифри цього числа були рівні значенню `k`.

bool is_base(int n, int k, int base) {

    int digit;

    // Перевіряємо кожен цифру числа в новій системі числення

    while (n > 0) {

        digit = n % base; // Знаходимо останню цифру числа в системі з
основою `base`

        if (digit != k){ // Якщо цифра не дорівнює `k`, то це не
підходить

            return false; // Повертаємо false, якщо хоч одна цифра
відрізняється від `k`

        }

        n /= base; // Ділимо число на основу, щоб перевірити
наступну цифру
```

```

    }

    return true; // Якщо всі цифри дорівнюють `k`, повертаємо true
}

// Функція знаходить мінімальну основу системи числення `base`,
// в якій число `n` складається тільки з цифр, рівних `k`
int find_base(int n, int k) {
    int base;

    // Починаємо цикл від основи, рівної `k`, до числа `n`
    for (base = k; base <= n; base++) {

        // Якщо число `n` дорівнює `k`, то шукаємо наступне число
        if (n == k) {
            return n + 1; // Повертаємо n+1, щоб уникнути некоректного
результату
        }

        // Використовуємо функцію is_base для перевірки, чи є `n`
числом в системі з основою `base`,
        // яке складається тільки з цифр `k`
        if (is_base(n, k, base)) {
            return base; // Якщо знайшли таку основу, повертаємо її
        }
    }

    return -1; // Якщо таку основу не знайдено, повертаємо -1
}

```

```

}

int main() {

    int n, k, x;

    // Введення значень `n` і `k` від користувача

    cin >> n >> k;

    // Викликаємо функцію find_base, щоб знайти мінімальну основу

    x = find_base(n, k);

    // Виводимо результат

    cout << x << endl;

    return 0;

}

```

**Функція `is_base`** перевіряє, чи можна подати число  $n$  в системі числення з основою `base`, щоб усі цифри були рівні значенню `k`.

**Функція `find_base`** шукає мінімальну основу системи числення, при якій число  $n$  можна записати лише з цифрами, рівними `k`.

**Основна програма** отримує від користувача два числа  $n$  та  $k$ , після чого обчислює та виводить мінімальну основу системи числення, яка задовольняє умови.

Завдання №3

1)  $y = 23, 2) x = 92$

3)  $y_2 = 010111$

$$\begin{array}{r} 23 \cdot 2 \\ 2 \overline{) 46} \\ \underline{4} \phantom{0} \\ 0 \phantom{0} \\ 0 \phantom{0} \\ 0 \phantom{0} \\ 0 \phantom{0} \end{array}$$

4)  $x_2 = 01011100$

$$\begin{array}{r} 92 \cdot 2 \\ 2 \overline{) 184} \\ \underline{184} \\ 0 \phantom{0} \\ 0 \phantom{0} \\ 0 \phantom{0} \\ 0 \phantom{0} \\ 0 \phantom{0} \end{array}$$

5)  $\begin{array}{r} 1011100 \\ 10111 \\ \hline 01110011 \end{array}$

6)  $\begin{array}{r} 1011100 \\ 10111 \\ \hline 01000101 \end{array}$

7)  $1011100 : 10111$

$$10111 = 10111 \cdot 10111 : 10111 = 1$$

$\frac{x_2}{y_2} = 0100$

8)  $\begin{array}{r} 1011100 \\ 10111 \\ \hline 1011100 \\ 1011100 \\ 1011100 \\ 1011100 \\ 1011100 \\ 1011100 \\ 1011100 \\ 1011100 \\ 1011100 \end{array}$

$x_2 \cdot y_2 = 0100001000100$

9)  $k = 46$

10)  $\begin{array}{r} 46 \cdot 4 \\ 4 \overline{) 184} \\ \underline{184} \\ 0 \end{array}$   $k_2 = 40$

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

Завдання №1 Деталі по виконанню і тестуванню програми

Перевіряємо за формулою в калькуляторі введенні тестові данні та перевіряємо, чи співпадають з результатом виконання програми



```
C:\Users\Lenovo\ai_programming_playground_2024\ai_12\anton_mykhalchuk\epic_1>a.exe
Enter principal amount of investment: 1234
Enter annual percentage rate: 23
Enter number of interest charges per year: 12
Enter time for which the money is invested, in years: 13
```

```
The total amount of the investment, given:
Principal amount of the investment 1234.00
The annual interest rate 23.00%
The number of interest accruals per year 12
and the time for which the money is invested 13 years
Will be equal: 23854.53
--
```

Підпис та №1 до блоку з виконанням та тестуванням програми

Час затрачений на виконання завдання: 1 год.

Завдання №2 Деталі по виконанню і тестуванню програми

Перевіряємо на різних випадках, чи спрацьовують умови:

Число дорівнює цифрі, тоді система числення повинна бути на одиничку більше

```
3 3
4
```

Переводимо будь-яке число  $n$  з однакових цифр  $k$  з будь-якої системи числення у десяткову.

Тоді у програму вводимо те число  $n$  у десятковій системі числення та цифру  $k$ .

Програма має вивести основу оригінальної системи числення з якої ми брали це число:

```
10 1
9
```

або може знайти меншу основу, яка підлягає умові задачі:

```
15 1
2
```

В прикладі вище 15 складається з одиничок в системі числення з основою 14, проте програма вивела основу 2, тому що в двійковій системі числення 15 теж складається з одиниць, а  $2 < 14$ .

Якщо не було знайдено жодної системи числення, де число складається з цифр k, то виводиться -1

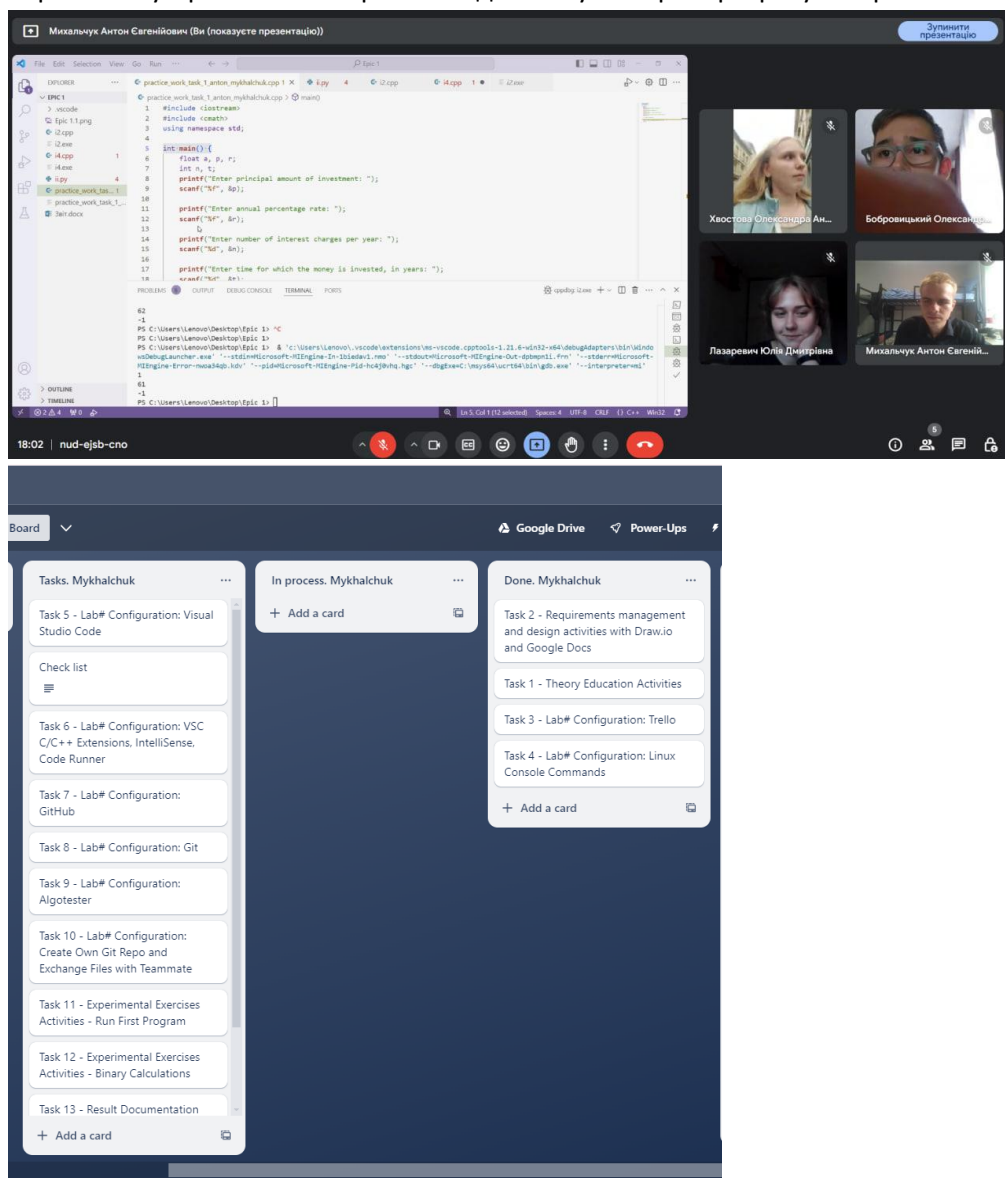
```
232 3
-1
```

Підпис та №2 до блоку з виконанням та тестуванням програми

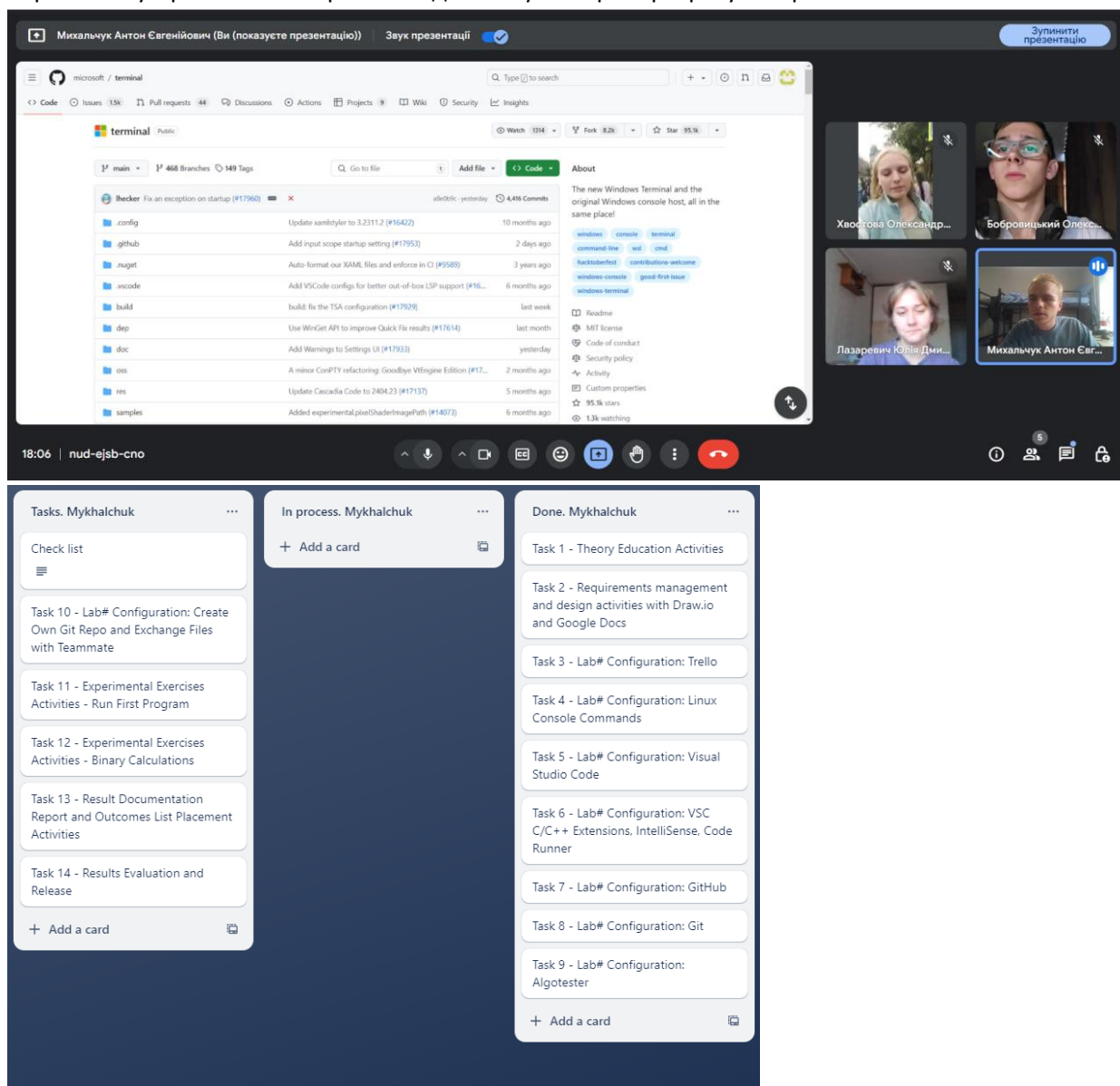
Час затрачений на виконання завдання: 2 год.

## 6. Кооперація з командою:

- Скрін з 1-ї зустрічі по обговоренню задач Епіку та Скрін прогресу по Тrello



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



- Скрін з 3-ї зустрічі по обговоренню задач Епіку та Скрін прогресу по Трелло (опційно)

Михальчук Антон Євгенійович (Ви покажете презентацію) Звук презентації

Зупинити презентацію

Trello Workspace

Recent Shared Templates Create

Google Drive Photos Up Automation Filters Share

Workspace studio My Board

Epics 1 Team 2

Your boards

Epics 1 Team 2

Done. Lazarenchuk

Tasks. Mykhailchuk

In process. Mykhailchuk

Done. Mykhailchuk

Tasks. Bidernytskyi

In process. Bidernytskyi

Хвостова Олександр...

Бобровицький Олекс...

Лазаревич Олія Дми...

Михальчук Антон Євг...

18:04 | nud-ejsb-cno

Tasks. Mykhailchuk

Check list

+ Add a card

In process. Mykhailchuk

+ Add a card

Done. Mykhailchuk

Task 1 - Theory Education Activities

Task 2 - Requirements management and design activities with Draw.io and Google Docs

Task 3 - Lab# Configuration: Trello

Task 4 - Lab# Configuration: Linux Console Commands

Task 5 - Lab# Configuration: Visual Studio Code

Task 6 - Lab# Configuration: VSC C/C++ Extensions, IntelliSense, Code Runner

Task 7 - Lab# Configuration: GitHub

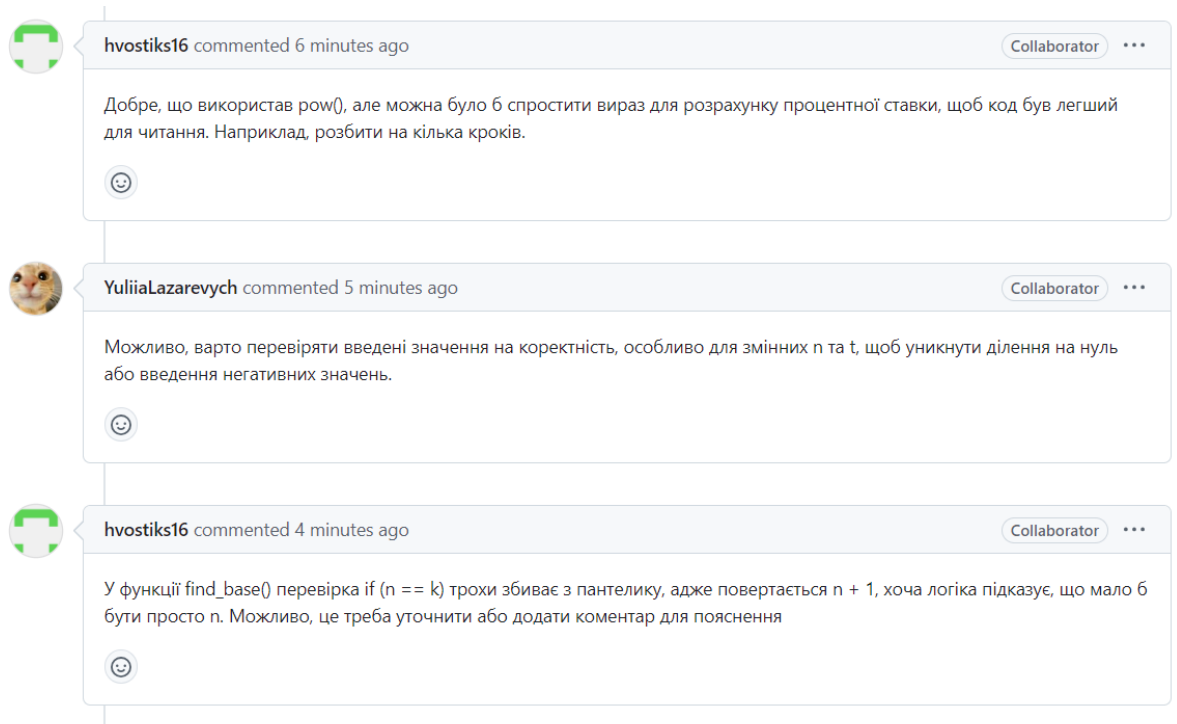
Task 8 - Lab# Configuration: Git

Task 9 - Lab# Configuration: Algotester

Task 10 - Lab# Configuration: Create Own Git Repo and Exchange Files with Teammate

+ Add a card

- Скрін з 2-му коментарями від учасників команди на пул реквесті з Ревю Роботи



## Висновки:

**Важливість Інструментів:** Використання сучасних інструментів, таких як пакетні менеджери та інтегровані середовища розробки, є критично важливим для ефективної роботи програміста. Вони не лише спрощують процес встановлення та налаштування програмного забезпечення, але також забезпечують потужні функції для оптимізації розробки коду.

**Розуміння Команд Терміналу:** Вивчення команд терміналу в Linux-орієнтованих середовищах допомагає розвинути навички роботи з командним рядком, що є основою для багатьох операцій у розробці та адмініструванні систем.

**Значення Дебагінгу:** Використання дебагера та лінера показує, як важливо виявляти помилки та покращувати якість коду. Завдяки цим інструментам стало можливим ефективно тестувати і вдосконалювати програми, що зменшує час на виправлення помилок.

**Структуроване Вивчення Програмування:** Вивчення основ програмування на C++ дозволяє створити базу знань, необхідну для розробки складних програм. Основи, такі як оголошення змінних, типи даних, оператори, умови та цикли, є критично важливими для формування навичок програмування.

**Практичний Досвід:** Виконання практичних завдань, таких як обчислення складних відсотків і робота з системами числення, демонструє застосування теоретичних знань у

реальних ситуаціях. Це сприяє глибшому розумінню концепцій програмування і дозволяє розвивати навички вирішення проблем.

**Співпраця в Команді:** Використання Git та GitHub показує, наскільки важливим є управління версіями у командній роботі. Ці інструменти дозволяють ефективно спілкуватися в команді, вести облік змін і спростити інтеграцію роботи кожного члена команди.