**Тема роботи**:

Основи програмування. Системи числення. Розробка середовища та перша програма.

**Мета роботи**:

Ознайомитися з процесом створення та виконання програм на C++, закріпити основи роботи з середовищем Visual Studio Code та системою контролю версій Git/GitHub. Використати для планування своїх дій Trello. Навчитися створювати прості алгоритми, відображати їх у вигляді блок-схем, працювати з двійковою системою числення та налагоджувати програмний код. Організувати роботу в команді.

**Теоретичні відомості:**

1. Теоретичні відомості з переліком важливих тем

- Тема №1: Програмування: алгоритм, програма, код.

- Тема №2: Середовище розробки (VS Code, Git, GitHub)

- Тема №3: Системи числення (десяткова, двійкова)

- Тема №4: Основи тестування та налагодження коду

1. Індивідуальний план опрацювання теорії:

-Тема №1: Програмування: алгоритм, програма, код.

* Джерела інформації
* Книжка: Kernighan & Ritchie “The C Programing Language”
* Відео: You Tube “C++ Tutorial for Beginners - Learn C++ in 1 Hour”
* Стаття: GeeksForGeeks – Algorithms basics
* Курс: Codecademy. Learn C++: Introduction
* Що опрацьовано:

Коментар 1: Прочитано базове визначення алгоритму

Коментар 2: Розібрано приклади простих програм

* Статус: ознайомлений
* Початок: 28.09.2025
* Завершення: 29.09.2025

-Тема №2: Середовище розробки (VS Code, Git, GitHub)

* Джерела інформації
* Книжка: Visual Studio Code by Bruce Johnson
* Відео: Git and GitHub Tutorial for Beginners
* Стаття: Visual Studio Code Docs
* Курс: "Introduction to Git and GitHub" Coursera
* Що опрацьовано:

Коментар 1: Встановлено VS Code та плагіни для C++

Коментар 2: Створено GitHub репозиторій і пул-реквест

* Статус: ознайомлений
* Початок: 28.09.2025
* Завершення: 29.09.2025

-Тема №3: Системи числення (десяткова, двійкова)

* Джерела інформації
* Книжка: С. М. Москвіна, Т. В. Грищук
* АЛГОРИТМІЧНІ МОВИ ТА ПРОГРАМУВАННЯ. ЕЛЕМЕНТИ ПРОФЕСІЙНОГО ПРОГРАМУВАННЯ
* Відео: An Introduction to Binary Numbers
* Стаття: Wikipedia – Binary numeral system
* Курс: Binary Studio Academy
* Що опрацьовано:  
  Коментар 1: Вивчено переведення чисел у двійкову систему  
  Коментар 2: Виконано приклади додавання у двійковій системі
* Статус: ознайомлений
* Початок: 29.09.2025
* Завершення: 29.09.2025

-Тема №4: Основи тестування та налагодження коду

* Джерела інформації
* Книжка: Steve McConnell “Code Complete”
* Відео: Debugging in VS Code (YouTube)
* Стаття: Medium – “Debugging tips for beginners”
* Курс: freeCodeCamp
* Що опрацьовано:  
  Коментар 1: Перевірено програму на відсутність синтаксичних помилок  
  Коментар 2: Змінено вивід для тестування
* Статус: ознайомлений
* Початок: 29.09.2025
* Завершення: 29.09.2025

**Виконання роботи:**

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

**Завдання №1**: Перша програма «Hello, World!»

**Варіант завдання**: Написати просту програму на C++, яка виводить повідомлення на екран.

**Деталі завдання**: Використати стандартну бібліотеку <iostream>, реалізувати функцію main(), додати інструкцію виводу cout.

**Важливі деталі для імплементації**: Синтаксис #include, використання namespace std, завершення програми оператором return 0.

**Завдання №2**: Модифікація виводу

**Варіант завдання**: Після рядка "Hello, World!" додати ще один власний рядок виводу.

**Деталі завдання**: Виконати редагування у файлі main.cpp.

**Важливі деталі для імплементації**: Використати окремий виклик cout для нового рядка, протестувати виконання у VS Code.

**Завдання №3**: Зарплата працівника

**Варіант завдання**: Програма обчислює заробітну плату працівника за кількістю відпрацьованих годин і погодинною ставкою.

**Деталі завдання**: Ввід: Ім’я працівника (рядок), Кількість годин (ціле число), Ставка (дійсне число). Вивід: Повідомлення у форматі: "Працівник <ім’я> отримав зарплату: <сума> грн"

Сума виводиться з точністю до 2 знаків після коми.

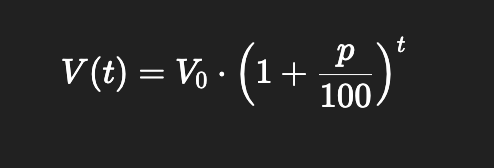
**Важливі деталі для імплементації**: Використання маніпулятора setprecision(2) для форматування дробових чисел. Правильне зчитування рядка (ім’я) разом із числовими значеннями. Тестування з різними значеннями (наприклад: великі години, ставка з дробовою частиною).

**Завдання №4:** Прогноз переглядів відео (TikTok / Instagram Reels)

**Варіант завдання:** Програма обчислює прогнозовану кількість переглядів відео через певну кількість днів при щоденному зростанні переглядів на p%.

**Деталі завдання:** Ввід: Початкова кількість переглядів Vo (дійсне число); Щоденний відсоток зростання p (дійсне число); Кількість днів t (ціле число). Вивід: Зростання у форматі з плаваючою точкою (2 знаки після коми) та прогнозована кількість переглядів через t днів (ціле число).

**Важливі деталі для імплементації:** Використання степеневої функції pow для обчислення формули:



Форматований вивід результатів з setprecision(2). Переведення результату в ціле число для зручного відображення прогнозу. Тестування з різними параметрами (різні відсотки росту, різна кількість днів).

**Завдання №5:** Двійкова система

**Варіант завдання:** Переведення чисел у двійкову систему та виконання операцій над ними.

**Деталі завдання:** Завдання виконане вручну в зошиті (розв’язки та обчислення оформлені письмово).

**Важливі деталі для імплементації:** Врахування правил переведення чисел у двійкову систему. Виконання арифметичних дій у двійковому вигляді. Остаточні розрахунки та перевірка правильності відповіді подані в зошиті.

**Завдання №6**: Practise Task (депозит, відсотки)

**Варіант завдання**: Реалізувати програму для обчислення складних відсотків за формулою:

IMG_256

**Деталі завдання**: Ввід даних від користувача (scanf): P, r (у %), t (роки), варіант виплати відсотків choice.

Переведення ставки з відсотків у десяткову: r = r / 100.0;.

Вибір періодичності нарахування:

choice == 1 → n = 12 (щомісячно)

choice == 2 → n = 4 (щокварталу)

інакше → n = 1 (щороку)

Обчислення: A = P \* pow((1 + r/n), (n\*t));

Вивід у форматі з 2 знаками після коми: вкладені кошти: P, загальна сума: A, заробіток (відсотки): A - P.

**Важливі деталі для імплементації**. Типи: double для P, r, A; int для n, t, choice. Форматований вивід (%.2f) для грошових сум. Бібліотеки: <cstdio>/<stdio.h>, <cmath> для pow.

**Завдання №7**: Algotester

**Варіант завдання:** Борщ, картопля і салат — потрібно мінімізувати сумарний ступінь недовіри, якщо всім видавати однакові порції.

**Деталі завдання:** Є n друзів, кожен хоче певну кількість борщу, картоплі та салату. Для всіх обираються порції A, B, C, і рахується сумарна різниця (|ai−A| + |bi−B| + |ci−C|). Потрібно знайти такі A, B, C, щоб сума була мінімальною.

**Важливі деталі для імлементації:** Використати медіану кожного масиву (борщ, картопля, салат). Для обчислення різниць застосувати abs(). Перевірка роботи коду на різних тестах у Algotester. Формат виходу: єдине ціле число — мінімальний сумарний ступінь недовіри.

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

Програма № 1: Перша програма «Hello, World!»

**Блок-схема**: Start → Вивести "Hello, World!" → End.

**Планований час на реалізацію**: 15 хв.

**Важливі деталі для врахування**: Коректне підключення бібліотеки <iostream>, перевірка роботи компілятора, запуск через Code Runner.

Програма № 2: Модифікація виводу

**Блок-схема**: Start → Вивести "Hello, World!" → Вивести додатковий рядок → End.

**Планований час на реалізацію**: 10 хв.

**Важливі деталі для врахування**: Використання кількох операторів cout, перевірка результату на консолі.

Програма №3: Зарплата працівника

**Блок-схема**: Start → Ввести ім’я, години, ставку → Обчислити зарплату = години × ставка → Вивести результат → End

**Планований час на реалізацію**: 25 хв.

**Важливі деталі для врахування**: Використання бібліотеки #include <iomanip>. . Використання формату double, float теж можна, але double більш точний. Якщо написати тільки setprecision(2), то це означає "2 значущі цифри", то потрібно додати fixed, щоб виводило саме 2 знаки після коми

Програма №4: Прогноз переглядів відео

**Блок-схема**: Start → Ввести Vo, p, t → Обчислити V = Vo × (1 + p/100)^t → Вивести результат (з округленням до двох знаків після коми) → End

**Планований час на реалізацію**: 35-40 хвилин

**Важливі деталі для врахування**: Використання бібліотеки #include <iomanip>. Використати pow() для степеня. Форматований вивід з точністю 2. Тестувати різні значення p (зростання/спад), t.

Програма №5: Practise Task (депозит, відсотки)

**Блок-схема**: Start → Ввести P, r, t → Обрати n (1/4/12) → Обчислити A за формулою → Вивести P, A, A–P → End

**Планований час на реалізацію**: ~60 хв

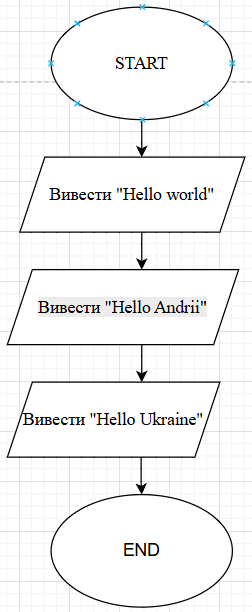
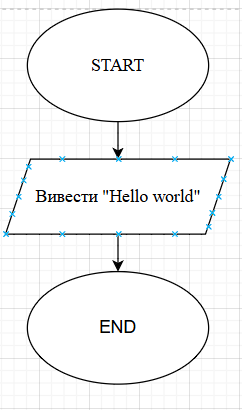
**Важливі деталі для врахування**: Коректність введення чисел (цілі та дробові). Форматований вивід результатів (%.2f). Тестування з різними параметрами (різні ставки, різний період, варіанти n).

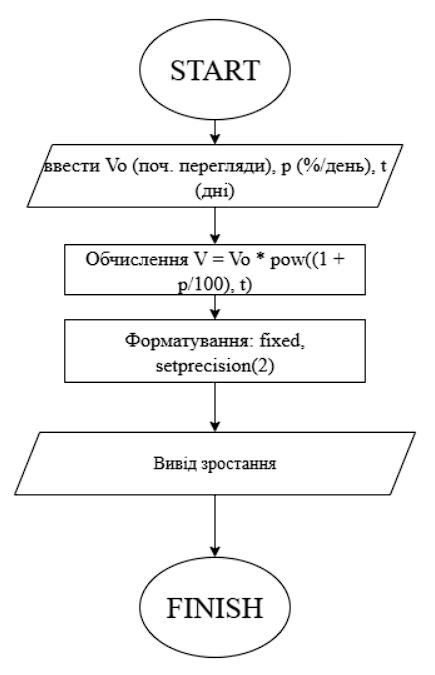
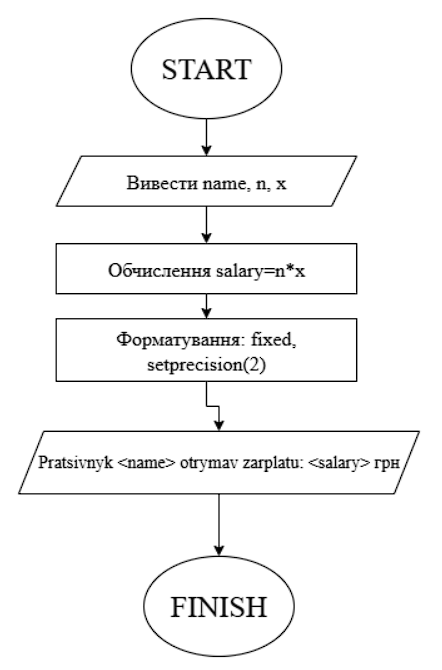
Програма №6: Algotester

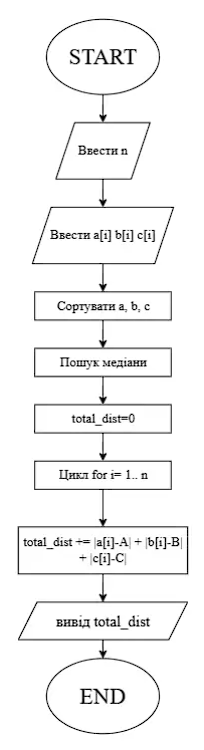
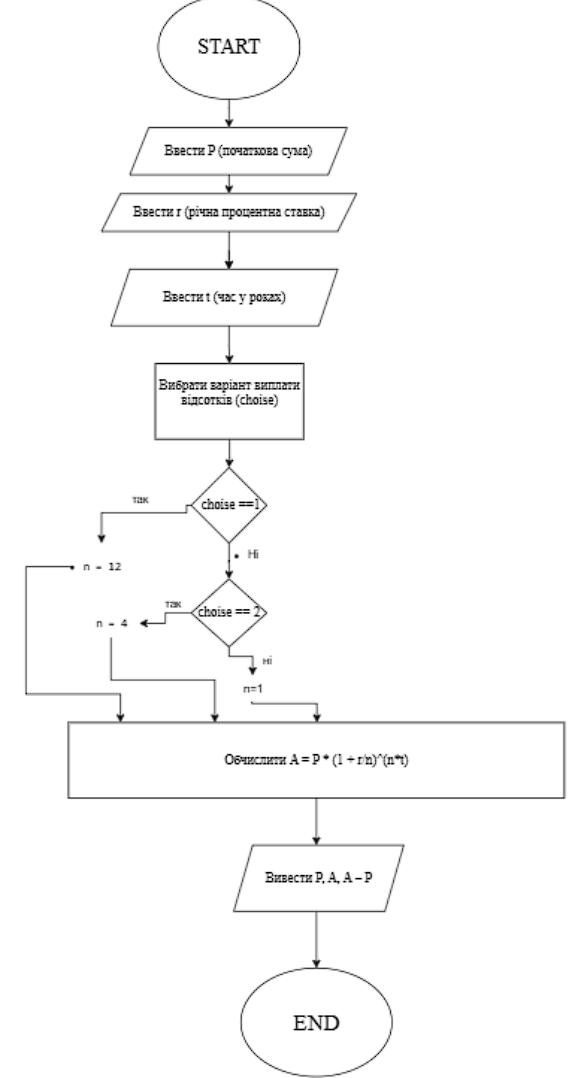
**Блок-схема:** Start → Ввести n → Ввести масиви **a[i]**, **b[i]**, **c[i]** →  
Відсортувати **a**, **b**, **c** → Обрати медіани: **A = median(a)**, **B = median(b)**, **C = median(c)** → **total\_dist = 0** → Цикл for i = 1..n: додати |a[i]−A| + |b[i]−B| + |c[i]−C| → Вивести **total\_dist** → End

**Планований час на реалізацію:** 30 хвилин

**Важливі деталі для врахування:** Використання **медіани** для мінімізації суми відхилень. Використовувати sort та abs для обчислень. Тестування на різних наборах даних (різна кількість друзів, різні значення).







**3. Конфігурація середовища до виконання завдань:**

**Завдання №1.** Встановлення Visual Studio Code.   
Використовується як основний редактор коду. Додатково встановлено розширення: C/C++ (для підтримки синтаксису та IntelliSense), Code Runner (для швидкого запуску програм), GitHub Pull Requests and Issues (для інтеграції з GitHub).

**Завдання №2**. Встановлення Git та підключення GitHub. Встановлено Git для роботи з репозиторіями. Виконано базову конфігурацію (git config --global user.name, git config --global user.email). Отримано доступ до командного рядка для виконання git-команд.

**Завдання №3**. Клонування репозиторію з GitHub

Виконано команду git clone <url> для завантаження репозиторію.

Створено окрему гілку для роботи (git switch -c andrii-epic1).

**Завдання №4**. Компілятор C++ (g++)

Використовується компілятор g++ для збирання і запуску програм.

Тестовий запуск показав коректну роботу.

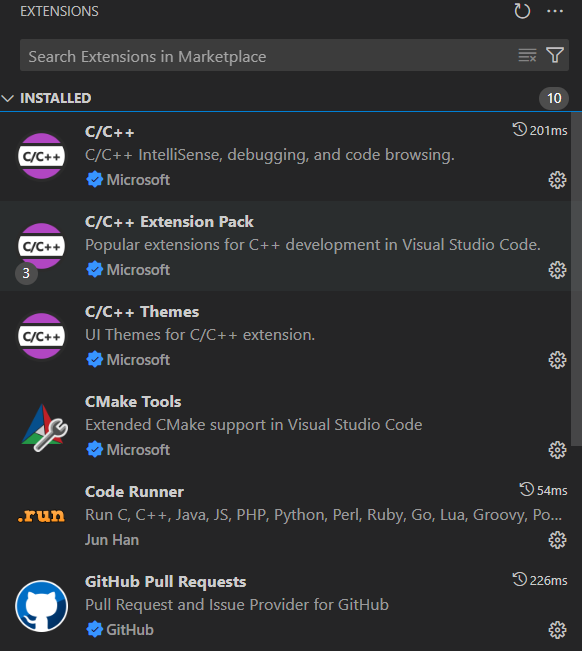


Рисунок 1. Встановлені розширення VS Code

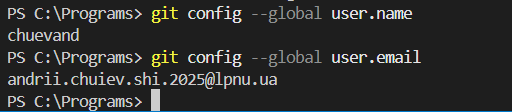


Рисунок 2. Конфігурація Git: налаштування імені користувача та електронної пошти.

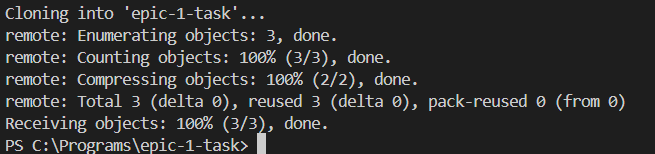


Рисунок 3. Клонування репозиторію з GitHub у локальне середовище.

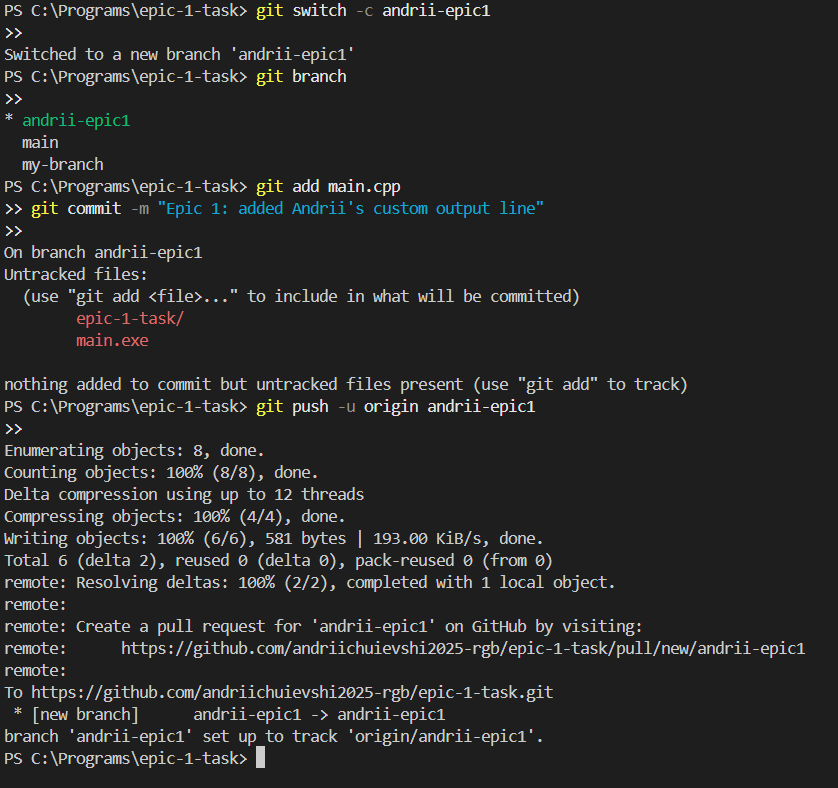


Рисунок 4Створення гілки, коміт змін та відправка гілки на GitHub.

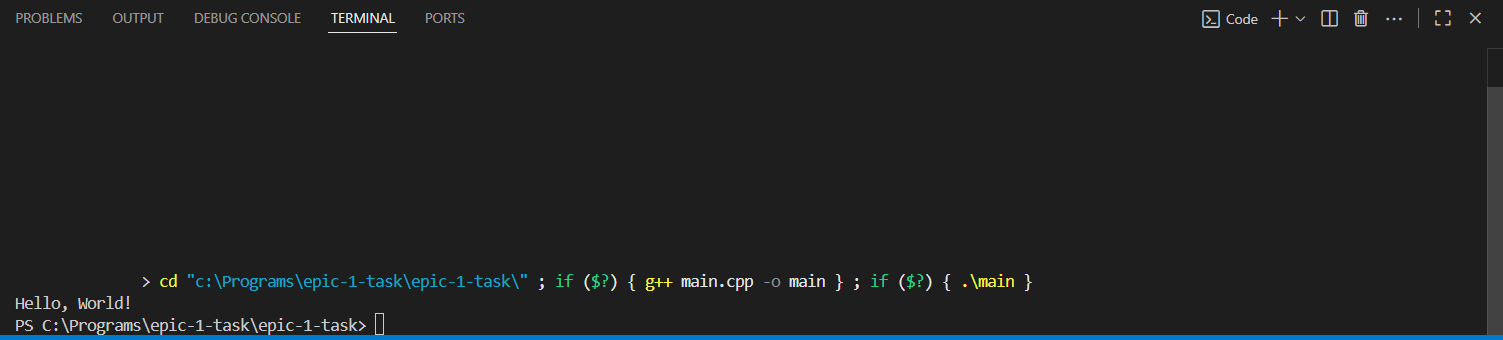


Рисунок 5. Перший запуск програми та перевірка її роботи.

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

#include <iostream>

int main(){

    std::cout << "Hello, World!" << std::endl;

    return 0;

}

Рисунок 6. Код програми “Hello world”

#include <stdio.h>

#include <iostream>

#include <iomanip>

using namespace std;

int main() {

    string name;

    int n;

    double x;

    cin >> name >> n >> x;

    cout<<"Pratsivnyk "<< name <<" otrymav zarplatu: "<< fixed << setprecision(2) <<x\*n << "grn" << endl;

return 0;

}

Рисунок 7. Код програми з обрахування зарплати працівника

#include <stdio.h>

#include <iostream>

#include <cmath>

#include <iomanip>

using namespace std;

int main() {

    double Vo, p, V;

    int t;

    cout << "Pochatkova kilkist perehliadiv: ";

    cin >> Vo;

    cout << "Zrostannia perehliadiv (%): ";

    cin >> p;

    cout << "Kilkist dniv: ";

    cin >> t;

    V=Vo\*pow((1+p/100),t);

    cout << "Zrostannia (float): " << fixed << setprecision(2) << V << endl;

    cout << "Prohnoz perehliadiv cherez " << t << " dniv: " << int(V) << endl;

    return 0;

}

Рисунок 8. Код програми з прогнозом зростання переглядів відео

#include <stdio.h>

#include <iostream>

#include <cmath>

int main() {

    double P,r, A;

    int n, t, choise;

    printf("Osnovna suma investytsii: ");

scanf("%lf",&P);

    printf("richna protsentna stavka: ");

    scanf("%lf",&r);

    r=r/100.0;

    printf("chas, na yakyi hroshi investuiutsia, u rokakh: ");

    scanf("%d",&t);

    printf("Oberit variant vyplaty vidsotkiv:\n");

    printf("1-kozhnoho misiatsia, 2-kozhnoho kvartalu, 3-kozhnoho roku\n");

    printf("Vash vubir: ");

    scanf("%d",&choise);

    if (choise==1) n=12;

    else if (choise==2) n=4;

    else n=1;

    A=P\*pow((1+r/n),(n\*t));

    printf("vkladeni koshty: %.2f grn\n",P);

    printf("zahalna suma investytsii: %.2f grn\n",A);

    printf("suma samoho zarobitku: %.2f grn\n",A-P);

    return 0;

}

Рисунок 9 . Обчислення складних відсотків (депозит)

#include <iostream>

using namespace std;

#include <vector>

#include <algorithm> //для сортування медіан

#include <cmath> //для abs

int main () {

    int n ; //кількість друзів

    cin >> n; //зчитуємо n

    vector <int> a (n),b(n),c(n); //створюємо вектор для зберігання кількості грамів борщу, картоплі і салату

    for (int i = 0; i < n; i++) { //створюємо цикл, який зчитує кожне замовлення від друзів

        cin >> a[i] >> b [i] >> c[i];

    }

    // сортуємо медіани

    sort(a.begin(), a.end());

    sort(b.begin(), b.end());

    sort(c.begin(), c.end());

    int A = a[n/2]; // медіана для борщу

    int B = b[n/2]; // медіана для картоплі

int C = c[n/2]; // медіана для салату

    // Обчислюємо сумарний ступінь недовіри

    long long total\_dist = 0;

    for (int i = 0; i < n; i++) {

        // Додаємо абсолютні відхилення для кожного друга

        total\_dist += abs(a[i] - A); // Недовіра від борщу

        total\_dist += abs(b[i] - B); // Недовіра від картоплі

        total\_dist += abs(c[i] - C); // Недовіра від салату

    }

    cout << total\_dist << endl;

    return 0;

}

Рисунок 10. Код для задачі з Алготестеру

Посилання на файл програми у пул-запиті GitHub: <https://github.com/olehpona-ai2025/epic-1-task/pull/8>

1. **Результати виконання завдань, тестування та фактично затрачений час:**

Завдання №1 Програма виводить на екран повідомлення "Hello, World!". Результат виконання підтверджує коректність роботи коду.

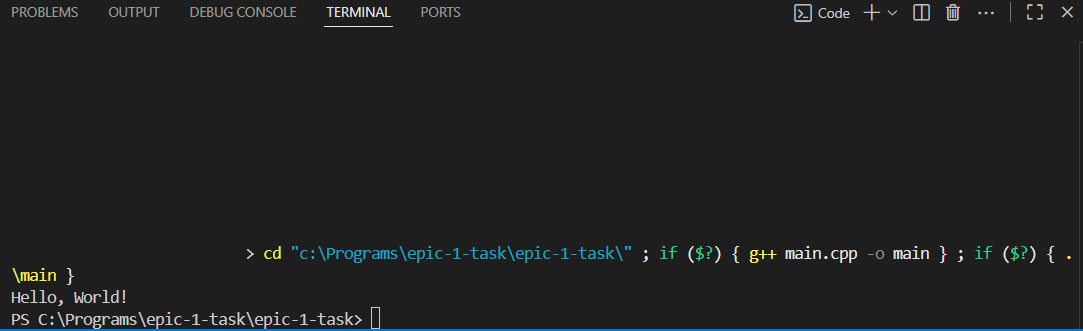


Рисунок 11. Тестування програми «Hello World».

Фактично затрачений час: 15 хв.

Завдання №2: Програма запитує ім’я працівника, кількість годин і погодинну ставку. Виводит на екран його зарплату.

Результати відповідають очікуваному алгоритму.

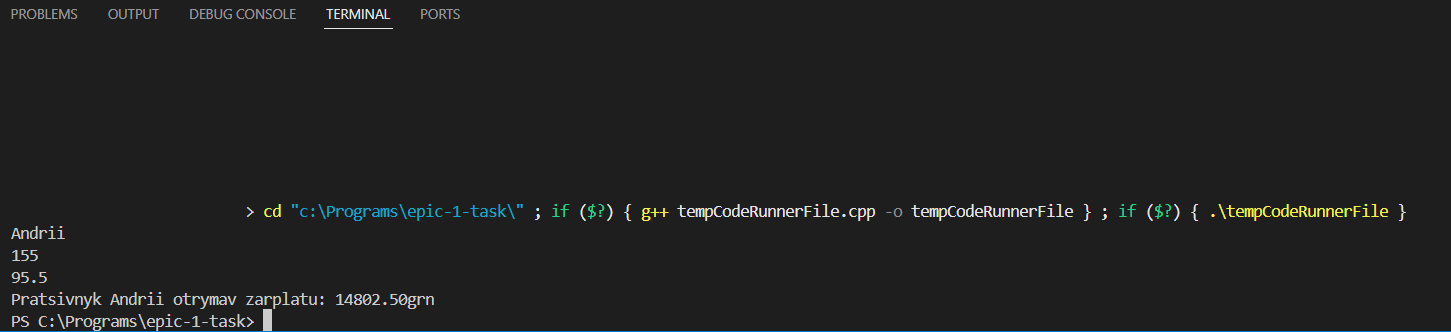


Рисунок 12. Тестування програми з обчисленням зарплати працівника.

Фактично затрачений час: 20 хв.

Завдання №3: Програма запитує початкову кількість переглядів, зростання переглядів у відсотках та кількість днів. На екрані виводиться зростання кількості переглядів днів упродовж встановленого часу (днів)

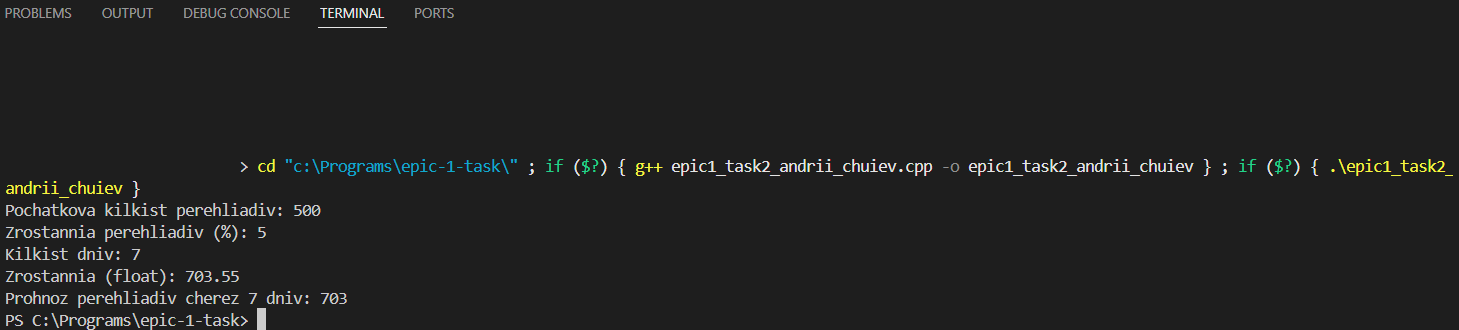


Рисунок 13. Тестування програми зі знаходженням зростання кількості переглядів

Фактично затрачений час: 40 хвилин.

Завдання №4: Програма повинна отримати значення від користувача (сума інвестиції, ставка, термін у роках, частота виплати відсотків), обчислити значення за формулою та вивести початкові й кінцеві дані.

Результати відповідають очікуваному алгоритму.

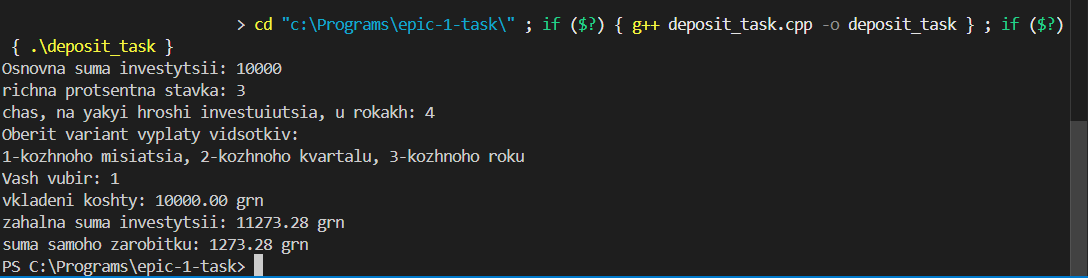


Рисунок 14. Тестування програми обчислення складних відсотків.  
Фактично затрачений час: 40 хв.

Завдання №5: Програма повинна мінімізувати сумарний ступінь недовіри всіх друзів, обравши такі порції борщу, картоплі й салату, які будуть однаковими для всіх. Для цього використовуються медіани значень бажаних порцій, що забезпечує мінімальне відхилення від потреб кожного друга.

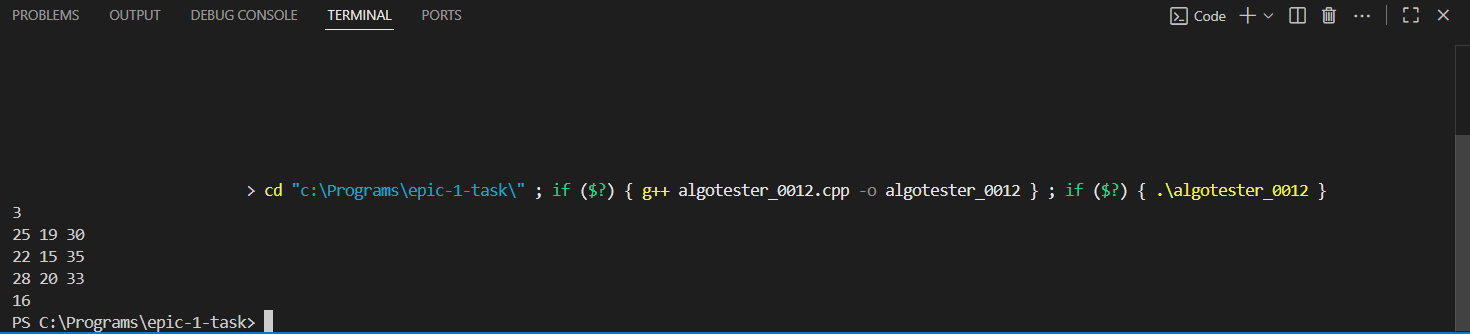
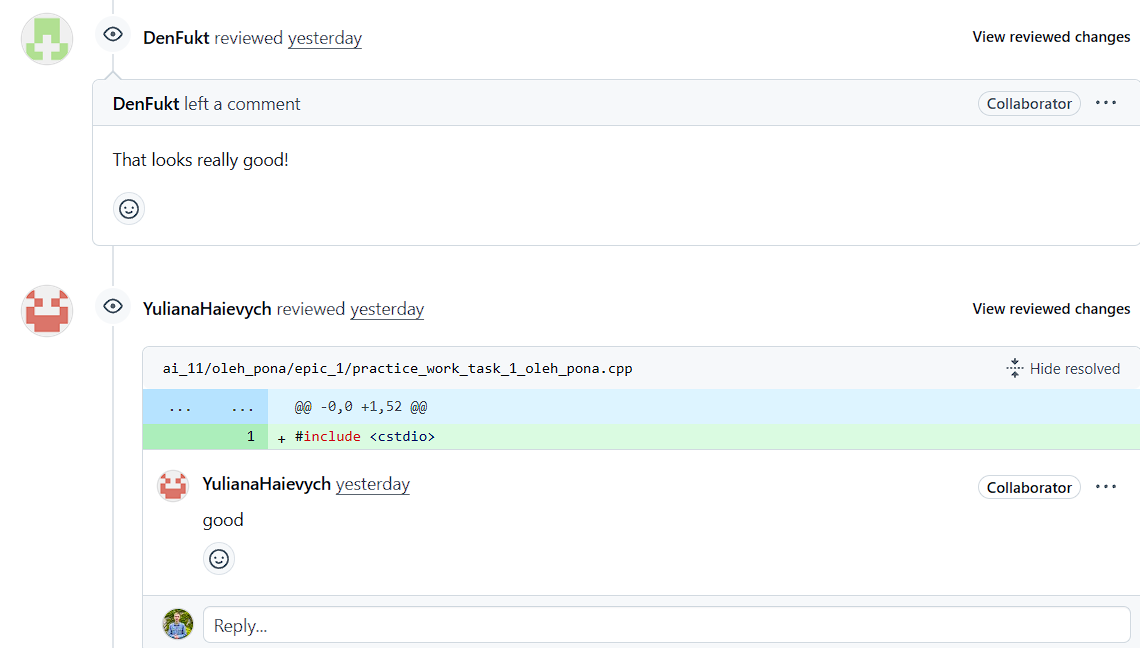


Рисунок 15. Тестування програми задачі з алготестеру

Фактично витрачений час: 30 хв.

**6. Кооперація з командою**  

Висновок: Ознайомився з процесом створення та виконання програм на C++, закріпив основи роботи з середовищем Visual Studio Code та системою контролю версій Git/GitHub. Використав для планування своїх дій Trello. Навчився створювати прості алгоритми, відображати їх у вигляді блок-схем, працювати з двійковою системою числення та налагоджувати програмний код. Організував роботу в команді.