

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

Виконала:

Студентка групи ШІ-11

Гуменюк Анастасія Олександрівна

Мета роботи: одержати практичні навички в розробці і дослідженні алгоритмів розв'язання задач.

Виконання роботи:

1) Опрацювання завдання та вимог до програми та середовища

Завдання №1 VNS Practice Work - Task 1

Варіант 16.
$$P = \frac{1 + \sin^2(x+1)}{2 + \left| x - 2x^3 / (1 + x^2 y^3) \right|} + x^4; \quad Q = \cos^2\left(\arctg \frac{1}{z}\right),$$

де $x=0,25$; $y=0,79$; $z=0,81$.

Завдання №2 VNS Practice Work - Task 2

Варіант 10.
$$y = \begin{cases} a^{b-x} + c, & x < 2, \\ b^{c-x} + a, & 2 \leq x \leq 4, \\ c^{a-x} + b, & x > 4; \end{cases}$$

де $x \in [0,5]$; $h_x = 0,31$; $a = 2$; $b = 3$; $c = 7$.

Завдання №3 VNS Practice Work - Task 3

Варіант 21. Обчислення опору електричного ланцюга, що складається з двох послідовно з'єднаних опорів. Нижче приведений вид екрану під час виконання програми, що рекомендується (дані, які вводяться користувачем, виділені напівжирним шрифтом). Обчислення опору електричного ланцюга при паралельному з'єднанні елементів.

Введіть початкові дані:

Величина першого опору (Ом) > **15**

Величина другого опору (Ом) > **27.3**

Опір ланцюга: 42.30 Ом

Реалізувати визначення суми Вашого внеску в банку «Альфа- Омега», якщо в кінці кожного року Ваш внесок збільшується не 3% від суми, що знаходиться на внеску (не від первинної суми, а від суми, що знаходиться на вкладі).

Програма повинна визначити суму Вашого внеску після 2-х перерахунків.

Первинний внесок – довільний і його значення повинне вводитися з клавіатури. Результат був виведений на екран монітора.

Завдання №4 VNS Practice Work - Task 4

Варіант 5. Написати програму, яка обчислює суму перших n цілих позитивних чисел. Кількість підсумованих чисел повинна вводитися під час роботи програми.

Завдання №5 Algotester_task_deadline

Дедлайн

Обмеження: 2 сек., 256 MiB

Одного дня Петрик задумався, що зовсім ніколи не встановлював дедлайнів на свої справи. Ніяких крайніх термінів, і так постійно! «Установлю собі дедлайн. Нарешті розпланую кожен день, щоб укластися в заплановану межу», — подумав Петрик. Своє наступне завдання з дедлайном Петрик розпочинає сьогодні.

Відомо, що сьогодні день тижня d , і дедлайн наступить через n днів.

Допоможіть Петрику, порахуйте кількість кожного дня тижня за наступні n днів, починаючи від сьогодні.

Вхідні дані

У першому рядку міститься назва сьогоднішнього дня тижня d .

У другому рядку міститься ціле число n .

Вихідні дані

У єдиному рядку виведіть 7 чисел — кількості кожного з днів тижня.

Перше число — це кількість понеділків, друге — кількість вівторків і т.д.

Обмеження

День d задається

рядком Monday (понеділок), Tuesday (вівторок), Wednesday (середа), Thursday (четвер), Friday (п'ятниця), Saturday (субота) або Sunday (неділя).

40% тестів: $1 \leq n \leq 10^5$,

60% тестів: $10^5 < n \leq 10^9$.

Завдання №6 Algotester_den_programista

День програміста

Обмеження: 2 сек., 256 MiB

Нарешті ми його дочекалися, 256-го дня в році (дня програміста)... Кожен зі студентів факультету прикладної математики та інформатики святкували його по-різному: дехто вдома, дехто в «Picasso», дехто в гуртожитку... Не дивно, що Зеник із Марічкою залишилися святкувати його в гуртожитку.

Вони організували «mega party» :-). Свято вдалося...

Наступного дня, гуляючи з Марічкою, Зенику стало цікаво, скільки ж було випито різної випивки?!

Марічка змогла згадати pn назв випивок, а Зеник — mn .

Вам потрібно написати програму, яка порахує скільки ж було різної випивки на святі, яку змогли згадати Зеник з Марічкою.

Вхідні дані

У першому рядку дано цілі числа n і m .

У наступних n рядках дано назви випивок, які змогла згадати Марічка.

В наступних m рядках дано назви випивок, які зміг згадати Зеник.

Вихідні дані

Єдине ціле число — кількість різної випивки на святі.

Обмеження

$$0 \leq n, m \leq 777$$

$$1 \leq |\text{name}_i| \leq 7 \text{ (лише маленькі латинські літери),}$$

$$|a| \text{ — довжина } a.$$

Завдання №7 Algotester_deputatski_kravatky

Депутатські краватки

Обмеження: 2 сек., 256 MiB

Колись давним-давно під час довгого нічного пленарного засідання Верховної Ради жіноча депутатська половина помітила, що не всі чоловіки-депутати прийшли у краватках. "Яке обурливе жахіття!" — подумали вони. Більше того, ті депутати-самці, що мали краватки, не узгодили їхню довжину. І це ми називаємо обличчям країни?!

Тож не гаючи часу, жіночки зібрали усі наявні краватки та вирішили змайструвати з них нові краватки для усіх чоловіків на засіданні. Їхня мета — краватки однакової довжини на усіх депутатах, що називають себе чоловіками. Для цього жінки можуть розрізати краватки ножицями довільну кількість разів. Поряд з цим депутатки хочуть зробити однакові нові краватки якомога довгими.

Тож Ваше завдання — допомогти жінкам знайти максимальну довжину!

Вхідні дані

У першому рядку задано два натуральні числа n та m — кількість зібраних краваток і кількість чоловіків-депутатів на пленарному засіданні відповідно.

У другому рядку задано n натуральних чисел a_i — довжина i -тої зібраної краватки.

Вихідні дані

У єдиному рядку виведіть одне число — максимальну довжину краватки.

Відповідь вважатиметься правильною, якщо її абсолютна чи відносна похибка не буде більшою ніж 10^{-4} .

Обмеження

$$1 \leq n, m \leq 1000,$$

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

Завдання №8 Algotester_pokemon

Назва для покемона

Обмеження: 2 сек., 256 MiB

Тарас знайшов новий вид покемонів, і тепер йому залишилося тільки придумати назву для нього. Ваше завдання — допомогти йому в цьому. Наразі у Тараса є рядок t , який складається із символів англійського алфавіту, а також знаків запитання. Для того, щоб завершити процес придумування назви, потрібно кожен знак запитання замінити на довільний символ. Крім цього відомо, що новий вид споріднений з іншим, давно відкритим видом під назвою s . Оскільки Тарас хоче, щоб назва нового виду відображала цю спорідненість, вона має містити s як підрядок.

Визначте, чи може Тарас замінити знаки запитання в назві t на символи таким чином, щоб відобразити спорідненість двох покемонів у назві.

Вхідні дані

У першому та другому рядках задано два рядки t та s .

Гарантується, що перший рядок складається з великих латинських літер та знаків запитання, а другий — лише з великих латинських літер.

Вихідні дані

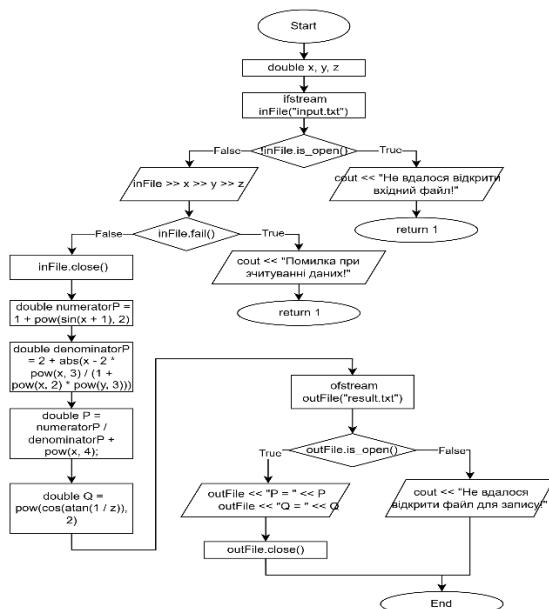
В єдиному рядку виведіть YES, якщо Тарас може придумати назву для нового покемона, або NO в протилежному разі.

Обмеження

$1 \leq |t|, |s| \leq 1000$.

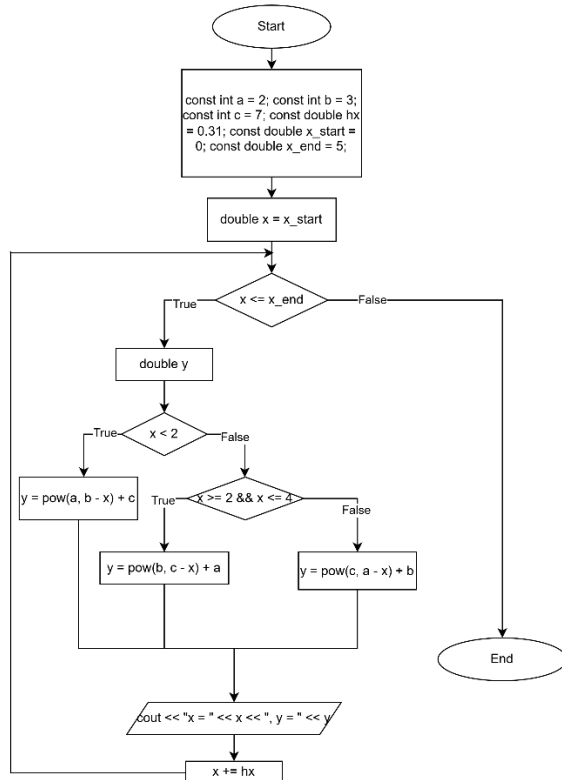
2) Дизайн виконання завдань:

Завдання №1 VNS Practice Work - Task 1



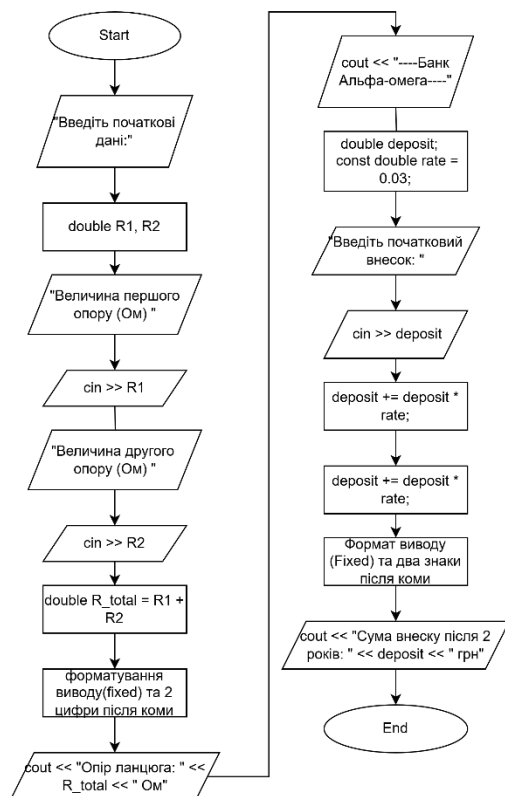
Планований час: 20 хв.

Завдання №2 VNS Practice Work - Task 2



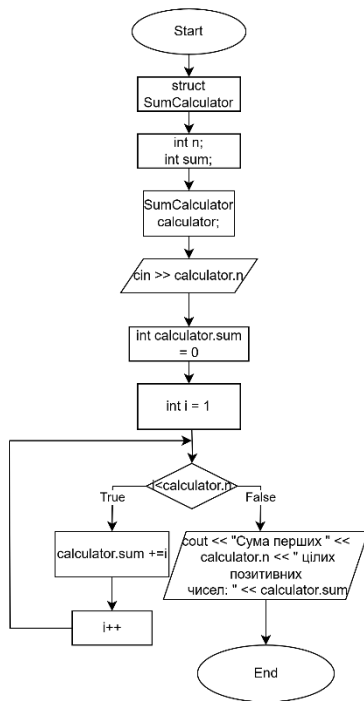
Планований час: 20 хв.

Завдання №3 VNS Practice Work - Task 3



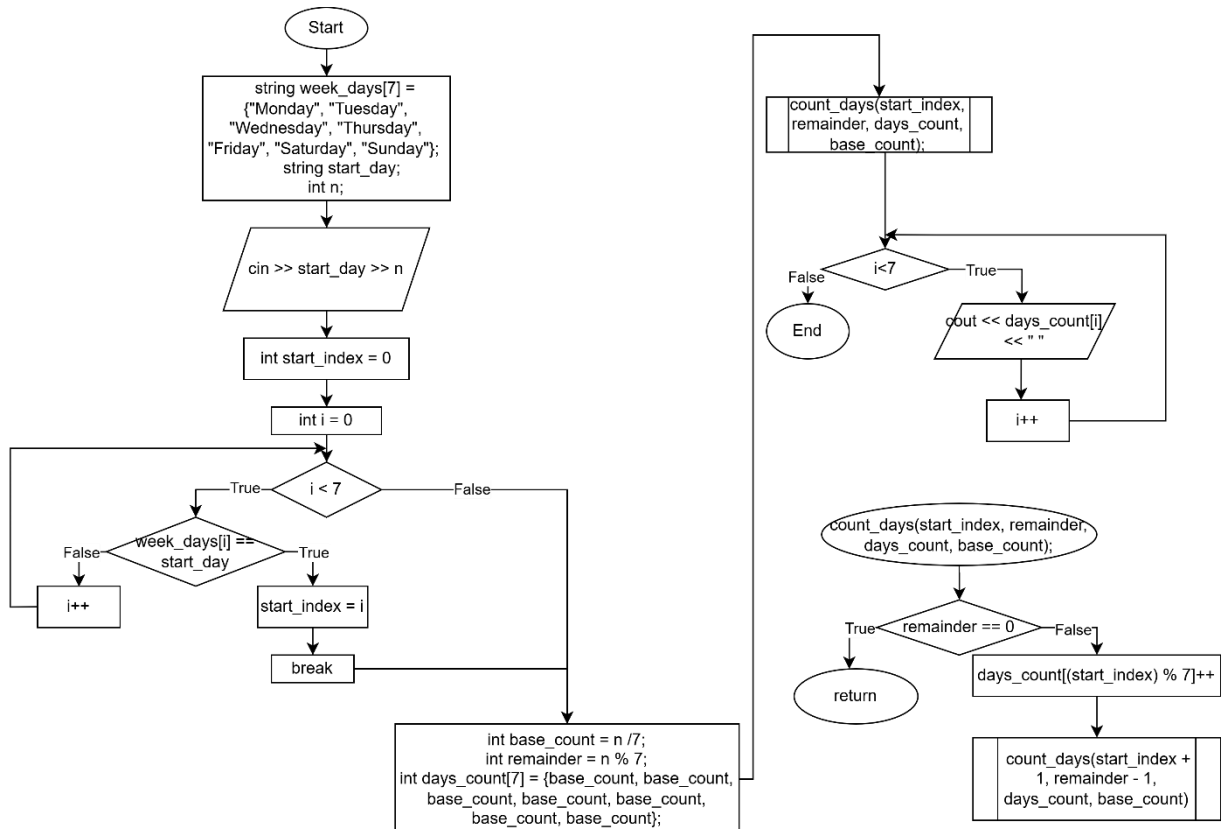
Планований час: 20 хв.

Завдання №4 VNS Practice Work - Task 4



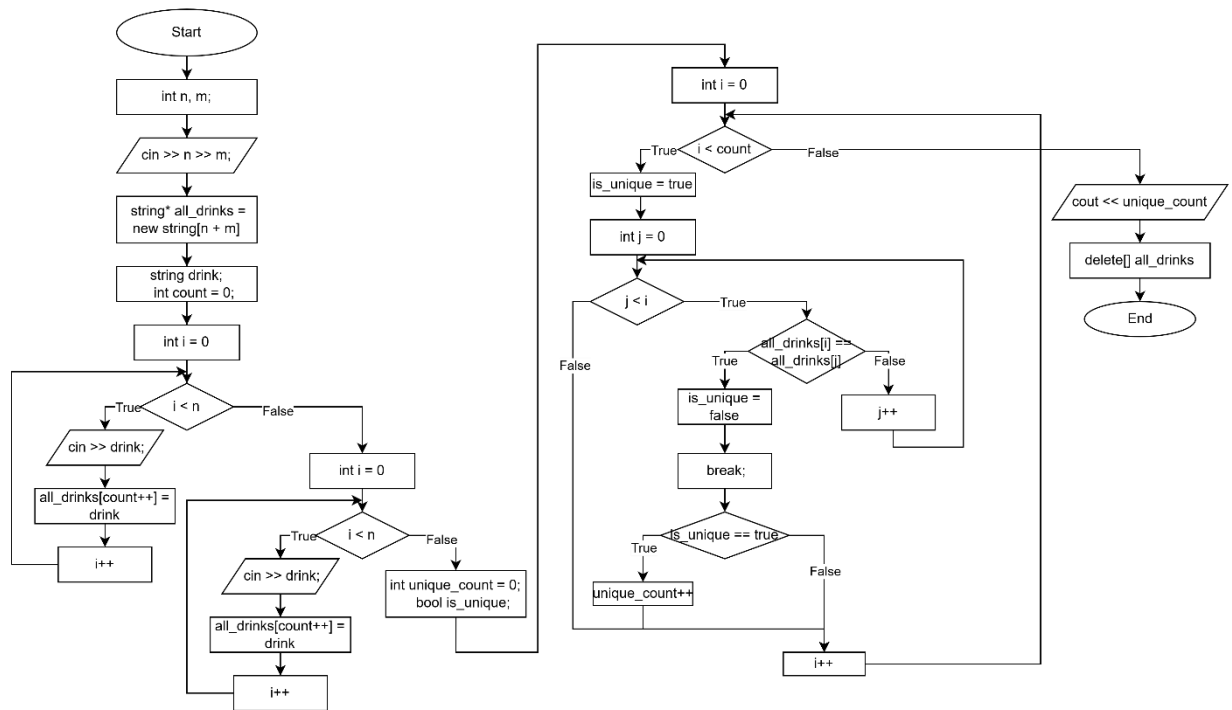
Планований час: 20 хв.

Завдання №5 Algotester_task_deadline



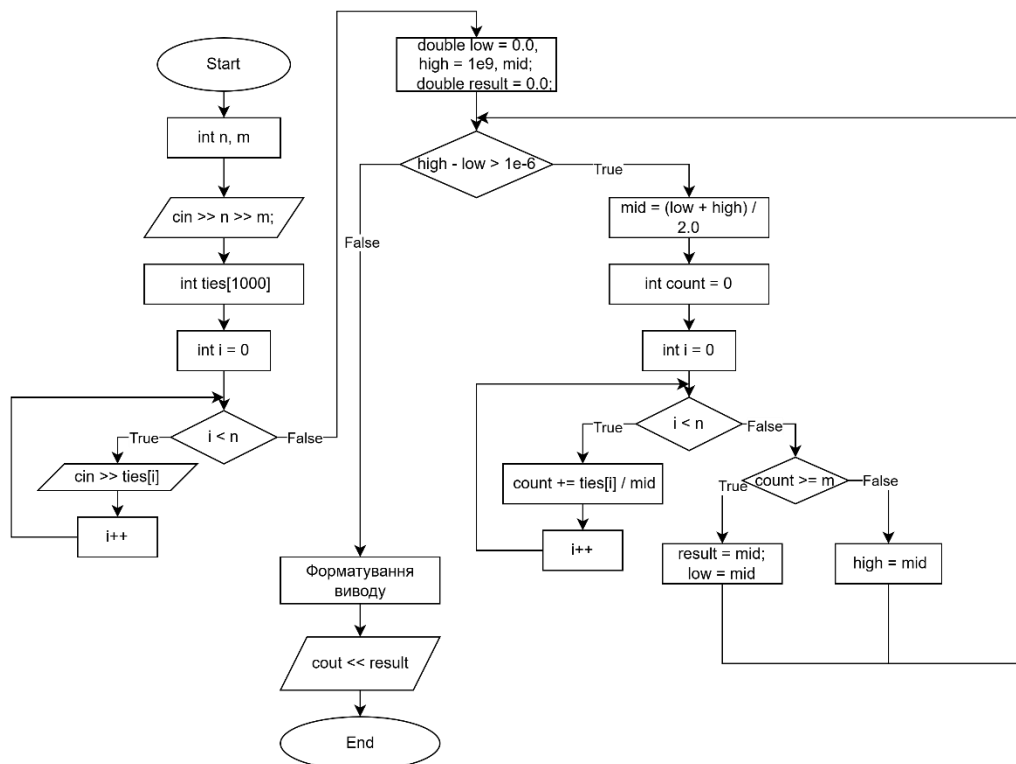
Планований час: 40 хв.

Завдання №6 Algotester_den_programista



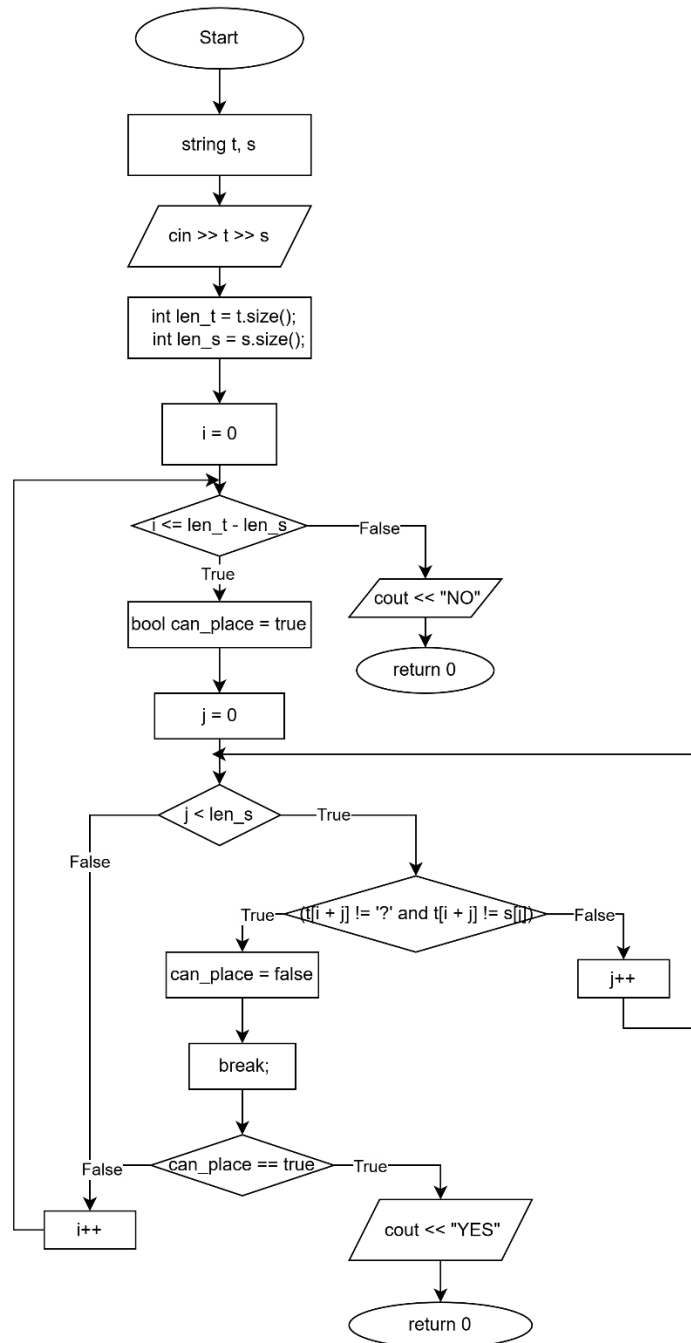
Планований час: 30 хв.

Завдання №7 Algotester_deputatski_kravatky



Планований час: 35 хв.

Завдання №8 Algotester_pokemon



Планований час: 30 хв.

3) Код програм з посиланням на зовнішні ресурси

Завдання №1 VNS Practice Work - Task 1

```
#include <iostream>
#include <fstream>
#include <cmath>

using namespace std;

int main() {
```

```

double x, y, z;

ifstream inFile("input.txt"); // Відкриття файлу для зчитування вхідних даних

if (!inFile.is_open()) {
    cout << "Не вдалося відкрити вхідний файл!" << endl;
    return 1;
}

inFile >> x >> y >> z; // Оператор введення даних

if (inFile.fail()) {
    cout << "Помилка при зчитуванні даних!" << endl;
    return 1;
}

inFile.close();

double numeratorP = 1 + pow(sin(x + 1), 2); // Математична операція
(піднесення до квадрату)
double denominatorP = 2 + abs(x - 2 * pow(x, 3) / (1 + pow(x, 2) * pow(y,
3))); // Математичні операції
double P = numeratorP / denominatorP + pow(x, 4); // Математична операція
(ділення, додавання, піднесення до ступеня)

double Q = pow(cos(atan(1 / z)), 2); // Математична операція (атангент,
косинус, піднесення до квадрату)

ofstream outFile("result.txt"); // Створення і відкриття файлу для запису
результатів

if (outFile.is_open()) {
    outFile << "P = " << P << endl; // Оператор виведення даних
    outFile << "Q = " << Q << endl;
    outFile.close();
} else {
    cout << "Не вдалося відкрити файл для запису!" << endl;
}

return 0;
}

```

Вимоги, дотримані у коді:

1. Дійсні змінні: Використано змінні x, y, z, numeratorP, denominator, P, Q, які є дійсними змінними типу double.
2. Математичні операції та функції: Використано математичні операції та функції: pow(), sin(), cos(), atan(), abs().

3. Функції роботи з файлами (зчитування): Використано функцію `ifstream` для зчитування значень з файлу `input.txt`.
4. Функції роботи з файлами (запис): Використано функцію `ofstream` для запису результату в файл `result.txt`.
5. Оператори введення та виведення даних: Використано оператори `cin` для введення даних та `cout` для виведення результату.
6. Умовні оператори: Використано оператор `if` для перевірки на успішне відкриття файлів і зчитування даних.

Завдання №2 VNS Practice Work - Task

```
#include <iostream>
#include <cmath>

using namespace std;

const int a = 2; // Використано константу
const int b = 3; // Використано константу
const int c = 7; // Використано константу
const double hx = 0.31; // Використано константу
const double x_start = 0; // Використано константу
const double x_end = 5; // Використано константу

int main() {

    for (double x = x_start; x <= x_end; x += hx) { // Використано цикл for
        double y;

        if (x < 2) { // Використано умовний оператор if
            y = pow(a, b - x) + c; // Математична операція
        } else if (x >= 2 && x <= 4) { // Використано умовний оператор else if
            y = pow(b, c - x) + a; // Математична операція
        } else {
            y = pow(c, a - x) + b; // Математична операція
        }

        cout << "x = " << x << ", y = " << y << endl; // Використано оператор
        виведення
    }

    return 0;
}
```

Вимоги, дотримані у коді:

1. Цілочисельні змінні: цілочисельні константи для a, b, c.
2. Константи: Використано константи для значень, які не змінюються під час виконання програми (наприклад, a, b, c).
3. Цикл for: Використано цикл for для ітерацій по значенням x від x_start до x_end з кроком hx.
4. Умовні оператори: Використано оператори if та else if для вибору формули для обчислення y.
5. Математичні операції: Використано математичні операції, зокрема піднесення до степеня та додавання.
6. Оператор виведення: Використано оператор cout для виведення результату обчислення.

Завдання №3 VNS Practice Work - Task 3

```
#include <iostream>
#include <iomanip>

using namespace std;

int main() {

    cout << "Введіть початкові дані:\n";

    double R1, R2;
    cout << "Величина першого опору (Ом) "; // Оператор виведення
    cin >> R1; // Оператор введення
    cout << "Величина другого опору (Ом) "; // Оператор виведення
    cin >> R2; // Оператор введення

    double R_total = R1 + R2; // Математична операція (додавання)

    cout << fixed << setprecision(2); // Форматування виведення для точності
    cout << "Опір ланцюга: " << R_total << " Ом" << endl; // Оператор виведення

    cout << "----Банк Альфа-омега----" << endl;

    double deposit; // Змінна для депозиту
    const double rate = 0.03; // Константа для процентної ставки

    cout << "Введіть початковий внесок: "; // Оператор виведення
    cin >> deposit; // Оператор введення

    deposit += deposit * rate; // 1 рік - Математична операція (обчислення відсотків)
```

```

    deposit += deposit * rate; // 2 рік - Математична операція (обчислення відсотків)

    cout << fixed << setprecision(2); // Форматування виведення для точності 2
    cout << "Сума внеску після 2 років: " << deposit << " грн" << endl; //
Оператор виведення

    return 0;
}

```

Вимоги, дотримані у коді:

1. Цілочисельна змінна: Використано змінні типу double для значень опорів і депозиту (R1, R2, deposit).
2. Дійсна змінна: Використано змінні типу double для математичних операцій і фінансових розрахунків (наприклад, R1, R2, R_total, deposit).
3. Цілочисельна константа: Використано константу rate для процентної ставки.
4. Математичні операції: Використано математичні операції для обчислення загального опору та процентів на депозиті.
5. Оператор виведення: Використано оператор cout для виведення результатів (наприклад, для опору ланцюга і суми депозиту).
6. Оператор введення: Використано оператори cin для введення значень користувачем (наприклад, для введення опорів та депозиту).
7. Форматування виведення: Використано setprecision для точності виведених значень (до двох знаків після коми).

Завдання №4 VNS Practice Work - Task 4

```

#include <iostream>
using namespace std;

// Оголошення структури для зберігання числа та суми
struct SumCalculator {
    int n;
    int sum;
};

int main() {
    SumCalculator calculator; //Створення об'єкта структури

    cout << "n = ";

```

```

cin >> calculator.n;

calculator.sum = 0;

int i = 1;

// Цикл do-while для обчислення суми перших n чисел
do {
    calculator.sum += i;
    i++;
} while (i <= calculator.n); // Перевірка умови продовження циклу

cout << "Сума перших " << calculator.n << " цілих позитивних чисел: " <<
calculator.sum << endl;

return 0;
}

```

Вимоги, дотримані у коді:

1. Цілочисельна змінна: Використано змінні n та sum типу int, де n зберігає кількість чисел, а sum — їх суму.
2. Структури: Використано структуру SumCalculator для зберігання значень n та sum.
3. Оператор введення: Використано оператор cin для введення значення n з клавіатури.
4. Цикл do-while: Використано цикл do-while для обчислення суми перших n цілих позитивних чисел.
5. Ініціалізація змінних: Ініціалізація змінної sum на нуль перед початком обчислення.
6. Оператор виведення: Використано оператор cout для виведення результату (сума чисел).
7. Оператор додавання: Використано оператор += для додавання числа до суми в кожній ітерації циклу.
8. Умови циклу: У циклі do-while умова продовження виконання перевіряється після кожної ітерації.

Завдання №5 Algotester_task_deadline

```

#include <iostream>
#include <string>

using namespace std;

// Використано рекурсивну функцію
void count_days(int start_index, int remainder, int days_count[], int base_count)
{ //Використовує параметри start_index, remainder, days_count[] i base_count
    if (remainder == 0) {

```

```

        return;
    }
    days_count[(start_index) % 7]++; // Математична операція (залишок від
ділення)
    count_days(start_index + 1, remainder - 1, days_count, base_count); //
Рекурсивний виклик
}

int main() {
    string week_days[7] = {"Monday", "Tuesday", "Wednesday", "Thursday",
"Friday", "Saturday", "Sunday"}; // Одновимірний масив для днів тижня

    string start_day;
    int n;

    cin >> start_day >> n; // Оператор введення даних

    int start_index = 0;
    for (int i = 0; i < 7; i++) { // Цикл for для пошуку індекса початкового дня
        if (week_days[i] == start_day) { // Умовний оператор (if) для перевірки
умови
            start_index = i;
            break; // Оператор break для примусового завершення циклу
        }
    }

    int base_count = n / 7; // Математична операція (цілочисельне ділення)
    int remainder = n % 7; // Математична операція (залишок від ділення)

    int days_count[7] = {base_count, base_count, base_count, base_count,
base_count, base_count, base_count}; // Одновимірний масив для кількості днів

    count_days(start_index, remainder, days_count, base_count); // Виклик
рекурсивної функції

    for (int i = 0; i < 7; ++i) { // Цикл for для виведення масиву
        cout << days_count[i] << " "; // Оператор виведення
    }
    cout << endl;

    return 0;
}

```

Вимоги, дотримані у коді:

1. Цілочисельна змінна: У коді використовуються змінні для зберігання цілих чисел, таких як кількість днів і індекси.
2. Умовні оператори та розгалуження: В коді є перевірки умов, що визначають виконання певних блоків коду.

3. Одновимірний масив: Використовуються масиви для зберігання певних значень, наприклад, днів тижня та кількості днів.
4. For цикл: Для перебору елементів застосовано цикл, що працює з заданим діапазоном.
5. Оператори break і continue: Присутні оператори для виходу з циклів та пропуску наступних ітерацій за певних умов.
6. Рекурсивна функція: Є функція, що викликає сама себе для виконання завдання.
7. Параметри та аргументи функції: Функція приймає кілька параметрів для обробки даних.
8. Математичні операції та математичні функції: Використовуються операції для математичних обчислень, такі як ділення та обчислення залишку.
9. Оператори введення та виведення даних: Застосовуються оператори для вводу і виводу даних.

Завдання №6 Algotester_den_programista

```
#include <iostream>
#include <string>

using namespace std;

int main() {

    int n, m, count = 0; // Використано цілочисельні змінні
    cin >> n >> m;

    string* all_drinks = new string[n + m]; // Використано вказівник для
динамічного масиву
    string drink;

    int i = 0;
    while (i < n) { // Цикл while для введення першої групи напоїв
        cin >> drink;
        all_drinks[count++] = drink;
        i++;
    }

    int j = 0;
    while (j < m) { // Цикл while для введення другої групи напоїв
        cin >> drink;
        all_drinks[count++] = drink;
    }
}
```



```

        j++;
    }

    // Використано цілочисельні змінні для підрахунку унікальних елементів
    int unique_count = 0;
    bool is_unique;

    int k = 0;
    while (k < count) { // Цикл while для перевірки унікальності напоїв
        is_unique = true;

        int l = 0;
        while (l < k) {

            if (all_drinks[k] == all_drinks[l]) { // Використано умовний оператор
if для порівняння елементів
                is_unique = false;
                break; // Оператор break для виходу з циклу
            }
            l++;
        }

        if (is_unique) {
            unique_count++;
        }
        k++;
    }

    cout << unique_count << endl;    // Оператор виведення даних

    delete[] all_drinks; // Звільнення пам'яті

    return 0;
}

```

Вимоги, дотримані у коді:

1. Цілочисельна змінна: Використано `int n, m, count`, які зберігають кількість напоїв та індекси.
2. Вказівники: Використано вказівник `all_drinks` для динамічного масиву рядків.
3. Цикл `while`: Використано два цикли `while` для введення напоїв та перевірки їх унікальності.
4. Умовні оператори: Використано оператор `if` для перевірки повторюваних напоїв.
5. Оператор `break`: Оператор `break` використано для виходу з циклу.
6. Оператор виведення: Використано оператор `cout` для виведення кількості унікальних напоїв.

Завдання №7 Algotester_deputatski_kravatky

```
#include <iostream>
#include <iomanip>
using namespace std;

int main() {

    int n, m; // Використано цілочисельні змінні
    cin >> n >> m;

    int ties[1000]; // Використано одновимірний масив для зберігання кількості
    краваток
    for (int i = 0; i < n; i++) {
        cin >> ties[i];
    }

    // Використано дійсні змінні для зберігання меж пошуку та результату
    double low = 0.0, high = 1e9, mid;
    double result = 0.0;

    while (high - low > 1e-6) { // Цикл while для бінарного пошуку
        mid = (low + high) / 2.0;

        int count = 0;

        for (int i = 0; i < n; i++) { // Цикл for для підрахунку кількості
    краваток, які можна отримати
            count += ties[i] / mid;
        }

        if (count >= m) { // Умовний оператор if для визначення меж бінарного
    пошуку
            result = mid;
            low = mid;
        } else {
            high = mid;
        }
    }

    cout << fixed << setprecision(7) << result << endl; // Оператор виведення з
    точною кількістю знаків після коми

    return 0;
}
```

Вимоги, дотримані у коді:

1. Цілочисельна змінна: Використано змінні `n`, `m` та `count` для зберігання кількості елементів та підрахунку.
2. Дійсні змінні: Використано змінні `low`, `high`, `mid`, `result` для бінарного пошуку та збереження результату.
3. Одновимірний масив: Масив `ties[1000]` для зберігання значень краваток.
4. Цикл `while`: Використано цикл `while` для бінарного пошуку.
5. Цикл `for`: Використано цикл `for` для підрахунку кількості краваток.
6. Умовний оператор: Використано оператор `if` для коригування меж бінарного пошуку.
7. Оператор виведення: Використано оператор `setprecision` для виведення результату з заданою точністю.

Завдання №8 Algotester_pokemon

```
#include <iostream>
#include <string>
using namespace std;

int main() {
    string t, s; // Використано цілочисельні змінні
    cin >> t >> s;

    int len_t = t.size(); // Використано цілочисельні змінні для довжини рядків
    int len_s = s.size();

    for (int i = 0; i <= len_t - len_s; i++) { // Цикл for для перебору всіх
        // можливих позицій у рядку t
        bool can_place = true;

        for (int j = 0; j < len_s; j++) { // Цикл for для перевірки відповідності
            // символів
            if (t[i + j] != '?' && t[i + j] != s[j]) { // Умовний оператор if для
                // перевірки символів
                can_place = false;
                break; // Оператор break для виходу з циклу
            }
        }

        if (can_place) { // Умовний оператор if для виведення результату
            cout << "YES" << endl;
            return 0;
        }
    }

    cout << "NO" << endl; // Оператор виведення результату
```

```
return 0;
}
```

Вимоги, дотримані у коді:

1. Цілочисельна змінна: Використано змінні `len_t` і `len_s` для зберігання довжин рядків.
2. Цикли `for`: Використано два цикли `for`: перший для перебору можливих індексів у рядку `t`, другий для порівняння символів з рядка `s`.
3. Умовний оператор `if`: Використано оператори `if` для перевірки умов співпадіння символів і для виведення результату.
4. Оператор `break`: Оператор `break` використано для виходу з циклу, якщо символи не співпали.
5. Оператор виведення: Використано оператори `cout` для виведення результату.

4) *Результати виконання завдань, тестування та фактично затрачений час:*

Завдання №1 VNS Practice Work - Task 1

input.txt	result.txt
1 0.25 0.79 0.81	1 P = 0.860141
	2 Q = 0.396172
	3

Фактичний час: 25 хв.

Завдання №2 VNS Practice Work - Task 2

```
PS C:\Users\User> & 'c:\Users\User\.vscode\extensions
=Microsoft-MIEngine-In-r04wirpw.1n1' '--stdout=Microso
soft-MIEngine-Pid-eoewtsi3.jvb' '--dbgExe=C:\msys64\u
x = 0, y = 15
x = 0.31, y = 13.4531
x = 0.62, y = 12.2054
x = 0.93, y = 11.1989
x = 1.24, y = 10.387
x = 1.55, y = 9.73208
x = 1.86, y = 9.20381
x = 2.17, y = 203.602
x = 2.48, y = 145.413
x = 2.79, y = 104.019
x = 3.1, y = 74.5726
x = 3.41, y = 53.6256
x = 3.72, y = 38.7247
x = 4.03, y = 3.01925
x = 4.34, y = 3.01053
x = 4.65, y = 3.00576
x = 4.96, y = 3.00315
PS C:\Users\User>
```

Фактичний час: 15 хв.

Завдання №3 VNS Practice Work - Task 3

```
PROBLEMS OUTPUT PORTS DEBUG CONSOLE TERMINAL

PS C:\Users\User> & 'c:\Users\User\.vscode\extensions\Microsoft-MIEngine-In-0socydb5.vro' '--stdout=Microsoft-MIEngine-Pid-qxpy50al.i23' '--dbgExe=C:\msys64\uc
Введіть початкові дані:
Величина першого опору (Ом) 15
Величина другого опору (Ом) 27.3
Опір ланцюга: 42.30 Ом
----Банк Альфа-омега----
Введіть початковий внесок: 1000
Сума внеску після 2 років: 1060.90 грн
PS C:\Users\User>
```

Фактичний час: 25 хв.

Завдання №4 VNS Practice Work - Task 4

```
PROBLEMS OUTPUT PORTS DEBUG CONSOLE TERMINAL

PS C:\Users\User> & 'c:\Users\User\.vscode\extensions\Microsoft-MIEngine-In-gbbmjrfk.kjt' '--stdout=Microsoft-MIEngine-Pid-2w5nm1nj.wxt' '--dbgExe=C:\msys64\uc
n = 4
Сума перших 4 цілих позитивних чисел: 10
PS C:\Users\User>
```

Фактичний час: 20 хв.

Завдання №5 Algotester_task_deadline

```
PROBLEMS OUTPUT PORTS DEBUG CONSOLE TERMINAL

PS C:\Users\User\Desktop\lpnu\epic5> & 'c:\Users\User\.vscode\extensions\Microsoft-MIEngine-In-xzktsp2n.hn4' '--stdin=Microsoft-MIEngine-Pid-4dxmunxz.ngf' '--dbgExe=C:\msys64\uc
Thursday
9
1 1 1 2 2 1 1
PS C:\Users\User\Desktop\lpnu\epic5>
```

Створено	Компілятор	Результат	Час (сек.)	Пам'ять (МіБ)	Дії
2 дні тому	C++ 23	Зараховано	0.002	1.188	Перегляд

Фактичний час: 30 хв.

Завдання №6 Algotester_den_programista

PROBLEMS OUTPUT PORTS DEBUG CONSOLE TERMINAL

```
PS C:\Users\User\Desktop\lpnu\epic5> & 'c:\Users\User\.vscode\extensions\ms-vscode.cpptools\bin\native\cl.exe' '--stdin=Microsoft-MIEngine-In-b2nepcyp.b5r' '--stdout=Microsoft-MIEngine-Out-b2nepcyp.b5r' '--pid=Microsoft-MIEngine-Pid-3dknsbus.5fz' '--dbgExe=C:\msys64\bin\gcc.exe'
1 2
medoff
tuborg
medoff
2
PS C:\Users\User\Desktop\lpnu\epic5> |
```

Створено	Компілятор	Результат	Час (сек.)	Пам'ять (МіБ)	Дії
2 дні тому	C++ 23	Зараховано	0.003	1.195	Перегляд

Showing 1 to 1 of 1 rows

Фактичний час: 25 хв.

Завдання №7 Algotester_deputatski_kravatky

PROBLEMS OUTPUT PORTS DEBUG CONSOLE TERMINAL

```
PS C:\Users\User\Desktop\lpnu\epic5> & 'c:\Users\User\.vscode\extensions\ms-vscode.cpptools\bin\native\cl.exe' '--stdin=Microsoft-MIEngine-In-4pg5d3f0.rr4' '--stdout=Microsoft-MIEngine-Out-4pg5d3f0.rr4' '--pid=Microsoft-MIEngine-Pid-tpysadwd.kke' '--dbgExe=C:\msys64\bin\gcc.exe'
3 4
1 10 5
3.3333327
PS C:\Users\User\Desktop\lpnu\epic5> |
```

Створено	Компілятор	Результат	Час (сек.)	Пам'ять (МіБ)	Дії
10 годин тому	C++ 23	Зараховано	0.003	1.441	Перегляд
10 годин тому	C++ 23	Зараховано	0.003	1.184	Перегляд
10 годин тому	C++ 23	Неправильна відповідь 1	0.002	0.922	Перегляд

Фактичний час: 40 хв.

Завдання №8 Algotester_pokemon

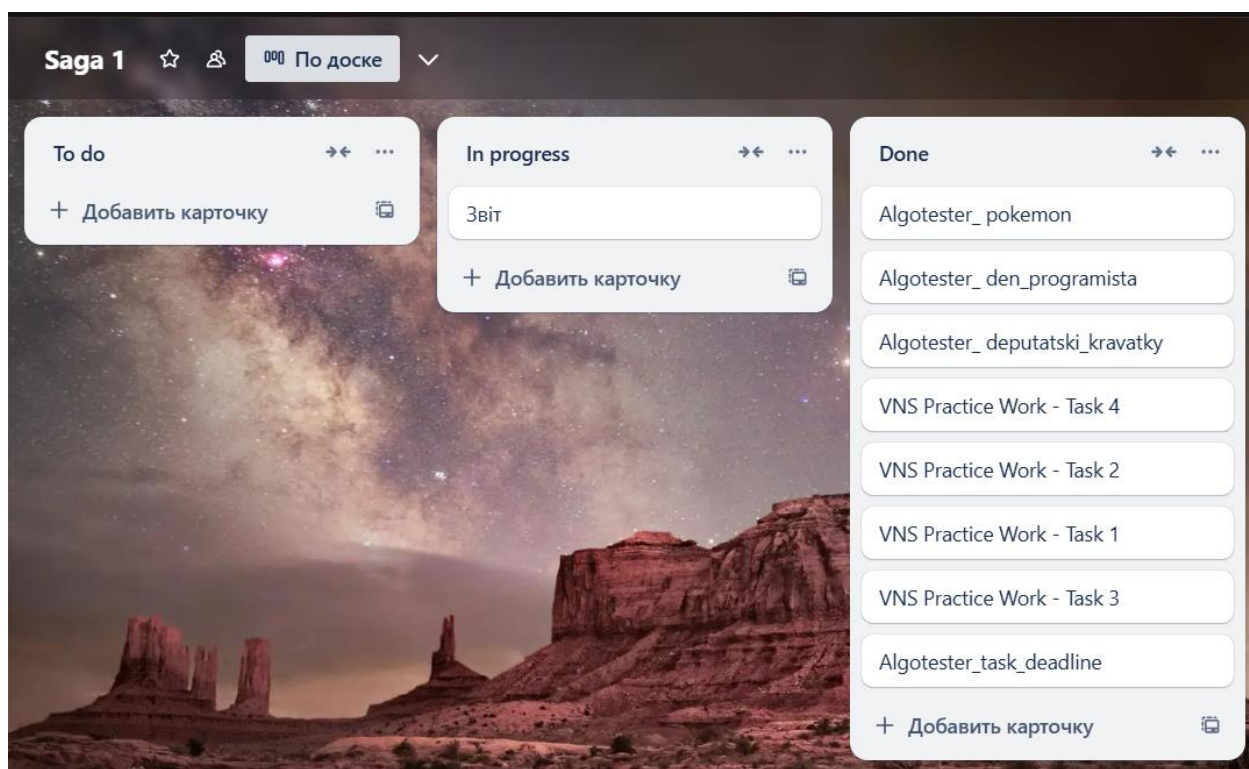
```
PROBLEMS OUTPUT PORTS DEBUG CONSOLE TERMINAL

PS C:\Users\User\Desktop\lpnu\epic5> & 'c:\Users\User\Desktop\lpnu\epic5\runner.exe' '--stdin=Microsoft-MIEngine-In-x351hnnd.4z
yr.soh' '--pid=Microsoft-MIEngine-Pid-gohlfdwr.yt1' '
PI??PI?KA??X
PIKACHU
YES
PS C:\Users\User\Desktop\lpnu\epic5> |
```

Створено	Компілятор	Результат	Час (сек.)	Пам'ять (МіБ)	Дії
11 годин тому	C++ 23	Зараховано	0.003	1.242	Перегляд

Фактичний час: 35 хв.

5) Дошка Trello для планування часу:



Висновок: виконавши розрахункову роботу, я здобула практичні навички в розробці та дослідженні алгоритмів для ефективного вирішення різноманітних задач. Це дозволило краще зрозуміти принципи алгоритмічного мислення та оптимізації рішень.