

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний аерокосмічний університет ім. М. Є. Жуковського
«Харківський авіаційний інститут»

Факультет систем управління літальних апаратів
Кафедра систем управління літальних апаратів

Лабораторна робота № 8

з дисципліни «Алгоритмізація та програмування»
на тему «Реалізація алгоритмів сортування та робота з файлами на мові C ++»

XAI.301. 141. 319a. 16ЛР

Виконав студент гр. 319a

Моїсеєнко Євген
(підпис, дата) (П.І.Б.)

Перевірив

 к.т.н., доц. Олена ГАВРИЛЕНКО
(підпис, дата) (П.І.Б.)

2023

МЕТА РОБОТИ

Вивчити теоретичний матеріал по алгоритмам обробки масивів на мові C++, а також бібліотеки для роботи з файлами і реалізувати оголошення, введення з файлу, обробку і виведення в файл одновимірних і двовимірних масивів на мові C++ в середовищі Visual Studio.

ПОСТАНОВКА ЗАДАЧІ

Завдання 1. За допомогою текстового редактору створити текстовий файл «array_in_n.txt» з елементами вихідного масиву (n - номер варіанта). У програмі на C++ перетворити масив відповідно до свого варіанту завдання (див. лаб.роб.№6, завд.2), ім'я файлу і необхідні змінні ввести з консолі. Вивести результати у файл «array_out_n.txt».

Array99.	Дан цілочисельний масив розміру N. Видалити з масиву всі елементи, що зустрічаються більше двох разів.
-----------------	--

Завдання 2. За допомогою текстового редактору створити текстовий файл «matr_in_n.txt» з елементами вихідного двовимірного масиву (n - номер варіанта). У програмі обробити матрицю відповідно до свого варіанту завдання (лаб.роб.№7, завд.1), ім'я файлу і необхідні змінні ввести з консолі. Дописати результати в той же файл.

Matrix36 °. Дана цілочисельна матриця розміру $M \times N$, елементи якої можуть набувати значень від 0 до 100. Різні рядки матриці назвемо схожими, якщо збігаються безлічі чисел, що зустрічаються в цих рядках. Знайти кількість рядків, схожих на перший рядок цієї матриці.

Завдання 3. Вивчити метод сортування відповідно до свого варіанту (див. табл. 1), проаналізувати його складність і продемонструвати на прикладі з 7-ми елементів (відповідно до свого варіанту). Реалізувати у вигляді окремої функції алгоритм сортування елементів масиву. Також окремими функціями реалізувати зчитування масиву з текстового файлу і виведення відсортованого масиву в консоль.

17	Двійкові вставки	Зростання	Дійсний
----	------------------	-----------	---------

ВИКОНАННЯ РОБОТИ

Завдання 1.

Вирішення задачі Array99

Процес читання та запису у файл відбувається у функції `main()`, яка в свою чергу є точкою входу в програму.

- `std::ifstream inputFile("C:\Users\Users\Desktop\array_in_99.txt");` В даному рядку коду створюється об'єкт файлового потоку для читання. Відкривається заданий файл.
 - `if (!inputFile.is_open()) {`
`std::cerr << "Не вдалося відкрити файл array_in_99.txt\n";`
`return 1; }` Даний блок коду перевіряє чи файл відкрився нормально та чи готовий до читання.
 - `inputFile >> N;` Читаємо з файлу ціле число значення кількості записаних чисел далі у файлі.
 - `std::vector<int> array(N);`
`for (int i = 0; i < N; ++i) {`
`inputFile >> array[i];`
`}` Читаємо з файлу масив цілих чисел
 - `inputFile.close();` Закриваємо файл
 - `removeDuplicates(array);` Видаляємо дублікати
 - `std::ofstream outputFile("C:\Users\Users\Desktop\array_out_99.txt");`
Створюється об'єкт файлового потоку для запису
 - `if (!outputFile.is_open()) {`
`std::cerr << "Не вдалося відкрити файл array_out_99.txt для запису\n";`
`return 1; }`
- Перевіряємо чи нормально відкрився файл.
- `outputFile << "Масив після видалення зайвих елементів: ";`
`for (int num : array) {`
`outputFile << num << " ";`

```
}
```

Записуємо масив у файл кожне значення через пробіл

- `outputFile.close();` Закриваємо файл

Лістинг коду вирішення задачі Array99 наведено в дод. А (7-10).

Екран роботи програми показаний на рис. Б.1

Завдання 2.

Вирішення задачі Matrix36

В даному коді процес читання з файлу відбувається в функції:

```
vector<vector<int>> readMatrixFromFile(const string& filename, int& M, int& N)
```

Функція отримує на вхід шлях до файлу та два посилання на цілі значення що будуть слугувати місцем збереження розмірів прочитаної матриці. Також функція в результаті виконання повертає вектор векторів типу INT що являє собою матрицю. В підсумку ми отримаємо матрицю та її розміри і зможемо опрацювати її як нам потрібно.

- `ifstream file(filename);` Відкриваємо вказаний файл

- ```
if (!file.is_open()) {
 cerr << "Помилка відкриття файлу.\n";
 exit(1);}
```

 Перевіряємо чи правильно відкрився файл

- ```
file >> M >> N;  
vector<vector<int>> matrix(M, vector<int>(N));
```

Читаємо розмір матриці з файлу та створюємо відповідний вектор векторів

- ```
for (int i = 0; i < M; ++i)
 for (int j = 0; j < N; ++j)
 file >> matrix[i][j];
```

Читаємо в циклі кожен рядок матриці

- `file.close();`

```
return matrix;
```

Закриваємо файл та повертаємо матрицю в точку виклику функції

Лістинг коду вирішення задачі Matrix36 наведено в дод. А (7-10).

Екран роботи програми показаний на рис. Б.2

Завдання 3

Вирішення задачі 17

Вхідні дані:

Ім'я файлу, що містить невпорядкований масив цілих чисел.

Опис: Файл містить невпорядкований масив цілих чисел, які розділені пробілами або новими рядками.

Тип: Рядок (string).

Обмеження: Назва файлу має відповідати існуючому файлу на системі, який містить невпорядкований масив цілих чисел.

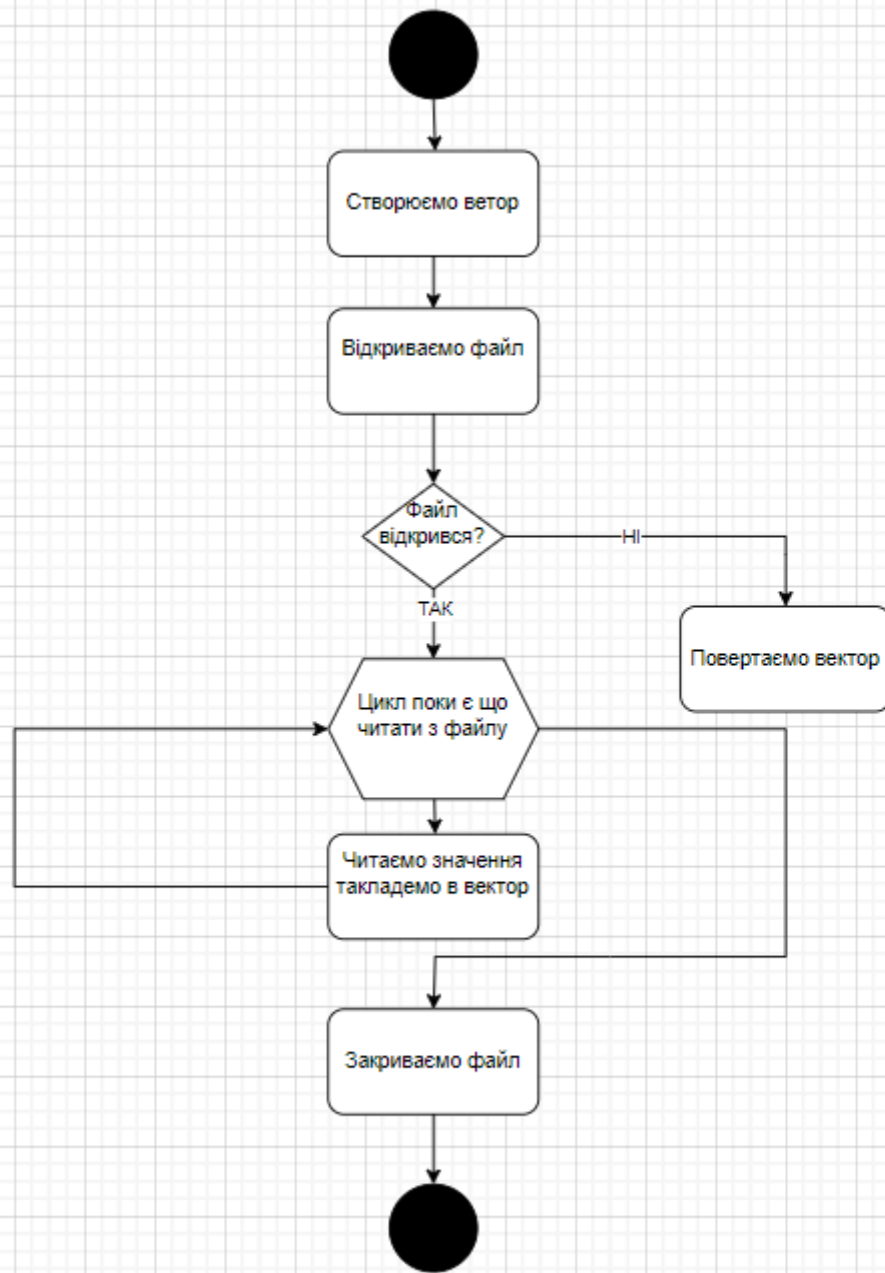
Вихідні дані:

Ім'я файлу, куди буде записаний впорядкований масив цілих чисел.

Опис: Файл, куди буде записаний впорядкований масив цілих чисел, розділених пробілами або новими рядками.

Тип: Рядок (string).

Алгоритм вирішення показано на рис. 17



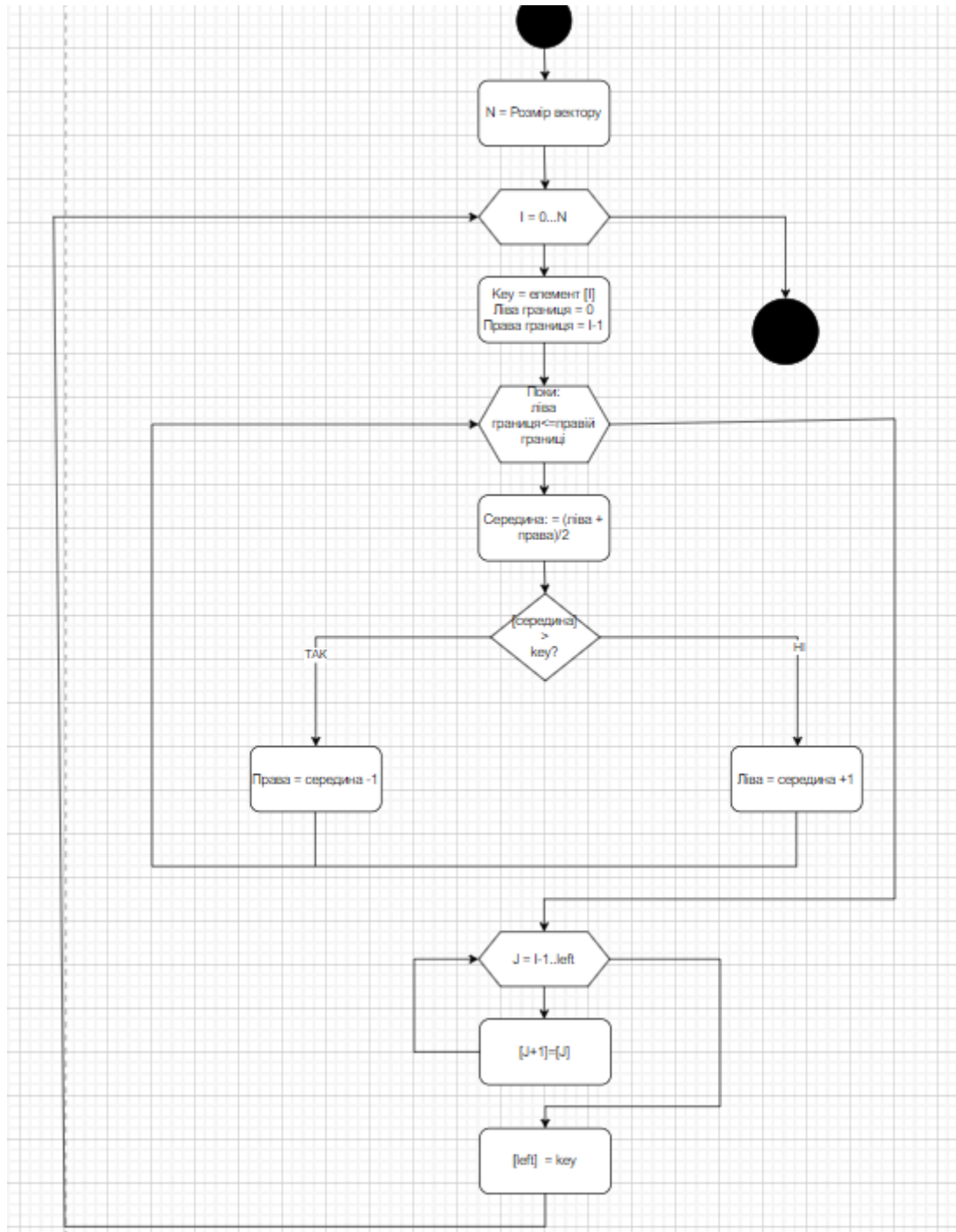


Рисунок 1 – 17

Лістинг коду вирішення задачі 17 наведено в дод. А (7-10).

Екран роботи програми показаний на рис. Б.3

## ВИСНОВКИ

У процесі вивчення теоретичного матеріалу по алгоритмах обробки масивів на мові C++ і роботи з файлами було здійснено реалізацію програмного коду в середовищі Visual Studio. Реалізовано функціонал введення даних з файлу, обробки одновимірних і двовимірних масивів за допомогою алгоритмів, а також виведення результатів у файл. Коротко кажучи, успішно опановано алгоритми обробки масивів та роботу з файлами у мові C++ в середовищі Visual Studio.



## ДОДАТОК А

### Лістинг коду програми

```
#include <iostream>
#include <fstream>
#include <vector>
#include <unordered_map>
#include <unordered_set>
#include <Windows.h>

using namespace std;

// Функція для видалення елементів з масиву, які зустрічаються більше двох разів
void removeDuplicates(vector<int>& arr) {
 unordered_map<int, int> frequency;
 vector<int> result;

 // Рахуємо кількість зустрічей кожного елемента
 for (int num : arr) {
 frequency[num]++;
 }

 // Заповнюємо результат лише елементами, що зустрічаються не більше двох разів
 for (int num : arr) {
 if (frequency[num] <= 2) {
 result.push_back(num);
 }
 }

 // Переприсвоюємо масив arr результатом
 arr = result;
}

// Функція для сортування масиву методом двійкового вставлення
void binaryInsertionSort(vector<int>& arr) {
 int n = arr.size();
 for (int i = 1; i < n; ++i) {
 int key = arr[i];
 int left = 0, right = i - 1;

 // Використовуємо бінарний пошук для знаходження позиції для вставки
 while (left <= right) {
 int mid = left + (right - left) / 2;
 if (arr[mid] > key)
 right = mid - 1;
 else
```

```

 left = mid + 1;
 }

 // Переміщуємо всі елементи більші за key на одну позицію вправо
 for (int j = i - 1; j >= left; --j)
 arr[j + 1] = arr[j];

 // Вставляємо елемент у відповідну позицію
 arr[left] = key;
}
}

// Функція для зчитування масиву з файлу
vector<int> readArrayFromFile(const string& filename) {
 vector<int> arr;
 ifstream inputFile(filename);
 if (!inputFile.is_open()) {
 cerr << "Unable to open file: " << filename << endl;
 return arr; // Повертаємо пустий масив у разі помилки
 }

 int number;
 while (inputFile >> number)
 arr.push_back(number);

 inputFile.close();
 return arr;
}

// Функція для виведення масиву у консоль
void printArray(const vector<int>& arr) {
 for (int num : arr)
 cout << num << " ";
 cout << endl;
}

// Функція для зчитування матриці з файлу
vector<vector<int>> readMatrixFromFile(const string& filename, int& M, int& N) {
 ifstream file(filename);
 if (!file.is_open()) {
 cerr << "Помилка відкриття файлу.\n";
 exit(1);
 }

 file >> M >> N;
 vector<vector<int>> matrix(M, vector<int>(N));

 for (int i = 0; i < M; ++i) {
 for (int j = 0; j < N; ++j) {
 file >> matrix[i][j];

```

```

 }
}

file.close();
return matrix;
}

// Функція для порівняння двох рядків матриці
bool areSimilar(const vector<int>& row1, const vector<int>& row2) {
 if (row1.size() != row2.size()) return false;

 unordered_set<int> set1(row1.begin(), row1.end());
 unordered_set<int> set2(row2.begin(), row2.end());

 return set1 == set2;
}

int main() {
 int choice;

 do {
 cout << "Меню:" << endl;
 cout << "1. Виконати програму для видалення зайвих елементів з масиву"
<< endl;
 cout << "2. Виконати програму для порівняння рядків матриці" << endl;
 cout << "3. Виконати програму для сортування масиву методом двійкового
вставлення" << endl;
 cout << "4. Вийти з програми" << endl;
 cout << "Виберіть опцію: ";
 cin >> choice;

 switch(choice) {
 case 1: {
 vector<int> myArray =
readArrayFromFile("C:\\Users\\Users\\Desktop\\in_17.txt");
 if (myArray.empty()) {
 cerr << "Array is empty or file could not be read." << endl;
 break;
 }
 cout << "Unsorted Array: ";
 printArray(myArray);
 removeDuplicates(myArray);
 cout << "Array after removing duplicates: ";
 printArray(myArray);
 break;
 }
 case 2: {
 int M, N;
 vector<vector<int>> matrix;

```

```

matrix =
readMatrixFromFile("C:\Users\Users\Desktop\math_in_36.txt", M, N);
 if (M == 0 || N == 0) {
 cout << "Матриця порожня.\n";
 break;
 }
 int similarCount = 0;
 for (int i = 1; i < M; ++i) {
 if (areSimilar(matrix[0], matrix[i])) {
 similarCount++;
 }
 }
 cout << "Кількість рядків, схожих на перший рядок: " <<
similarCount << "\n";
 break;
}
case 3: {
vector<int> myArray =
readArrayFromFile("C:\Users\Users\Desktop\in_17.txt");
 if (myArray.empty()) {
 cerr << "Array is empty or file could not be read." << endl;
 break;
 }
 cout << "Unsorted Array: ";
 printArray(myArray);
 binaryInsertionSort(myArray);
 cout << "Sorted Array: ";
 printArray(myArray);
 break;
}
case 4:
 cout << "До побачення!" << endl;
 break;
default:
 cout << "Невірний вибір. Будь ласка, виберіть знову." << endl;
}
} while (choice != 4);

return 0;
}

```

## ДОДАТОК Б

### Скрін-шоти вікна виконання програми

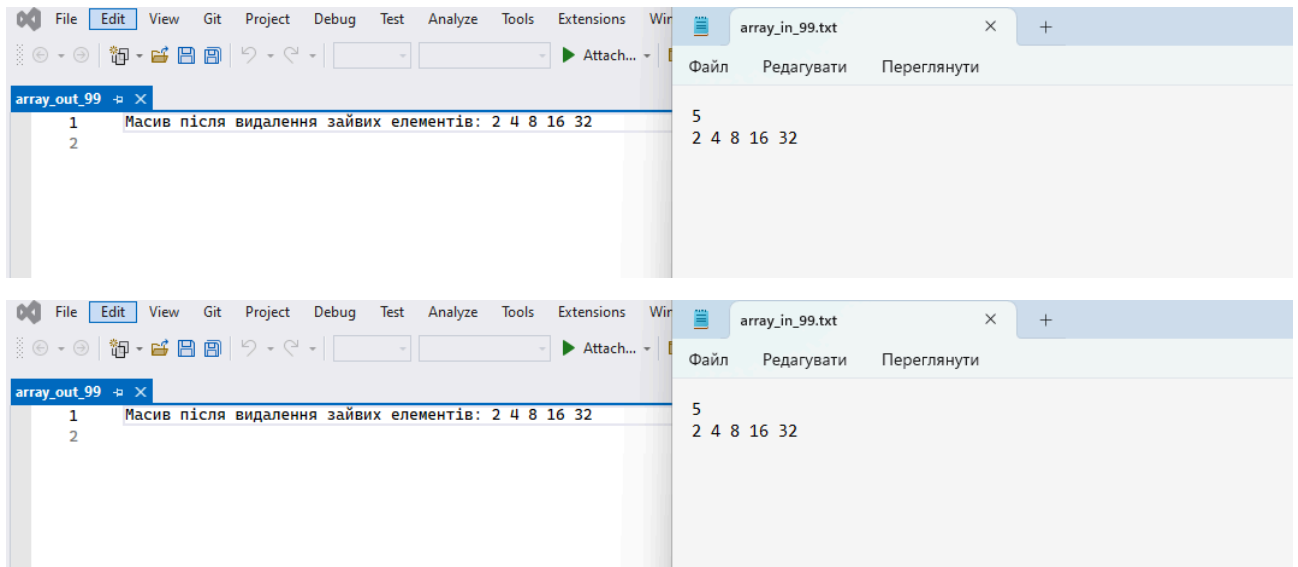


Рисунок Б.1 – Екран виконання програми для вирішення завдання Array99

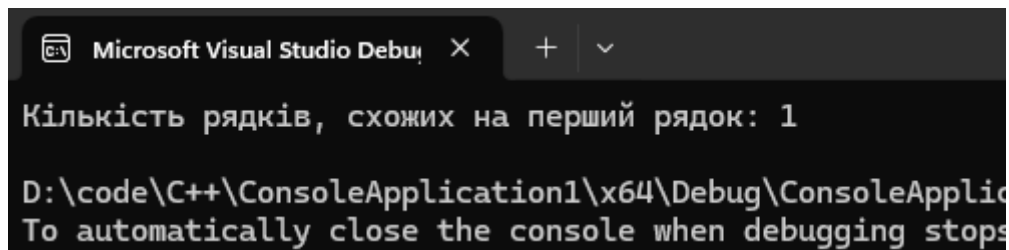


Рисунок Б.2 – Екран виконання програми для вирішення завдання Matrix36

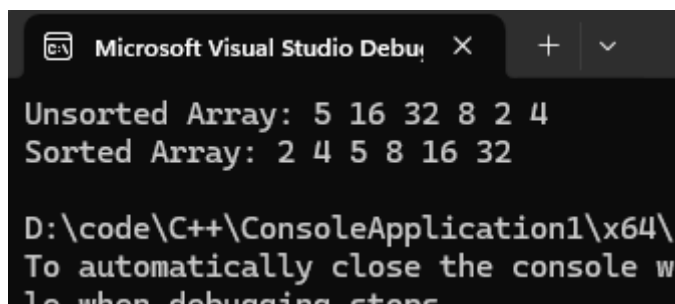
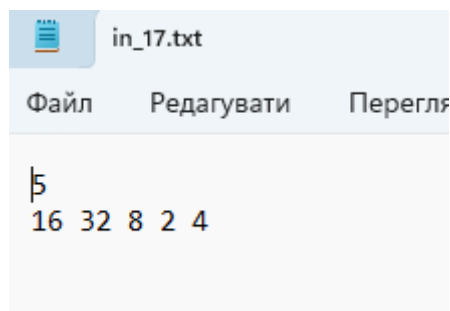


Рисунок Б.3 – Екран виконання програми для вирішення завдання