МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Одеський національний університет імені І. І. Мечникова

Факультет математики, фізики й інформаційних технологій

Кафедра комп'ютерних систем та технологій

КУРСОВА РОБОТА

з дисципліни

«Архітектура комп’ютерів»на тему:

«РЕАЛІЗАЦІЯ ЕМУЛЯТОРА НАВЧАЛЬНОЇ МАШИНИ»

Варіант 11

Студента ІІ курсу 1 групи
спеціальності «Комп’ютерна інженерія»
Окучинський Г.П.

Керівник: старший викладач
Берков Ю. М.

Одеса-2023

ЗМІСТ

ВСТУП.....1
СИСТЕМА КОМАНД……..…….……....2
Команда пересилання…….…4
Команда додавання дійсних чисел…….…..4
Команда віднімання дійсних чисел……..4
Команда множення дійсних чисел……4
Команда ділення дійсних чисел…….……5
Команда введення масиву дійсних чисел……5
Команда введення масиву цілих чисел……..…….5
Команда безумовного переходу……..5
Команда переведення дійсного числа в ціле……5
Команда додавання цілих чисел……6
Команда віднімання цілих чисел……6
Команда множення цілих чисел……6
Команда ділення цілих чисел……..6
Команда виводу масиву дійсних чисел……6
Команда виводу масиву цілих чисел……7
Команда умовного переходу(залежна від прапора нульового результату)……...7
Команда умовного переходу (залежна від прапора ω)……...…8
Команда переведення цілого числа в дійсне……8
Команда остачі від ділення цілих чисел……8
Команда ітерації масивом……9
Команда зупинки виконання програми……9
РЕАЛІЗАЦІЯ НАВЧАЛЬНОЇ МАШИНИ……..…10
ПРОГРАМИ ДЛЯ НАВЧАЛЬНОЇ МАШИНИ……..13
ЛІНІЙНІ ОБЧИСЛЕННЯ……..…13
ОБЧИСЛЕННЯ ЗА ДОПОМОГОЮ ЦИКЛІВ…….…16
ОБЧИСЛЕННЯ З ВИКОРИСТАННЯМ ЦИКЛІВ (МАСИВИ)……21
ВИСНОВКИ……..24
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ……25
Додаток А……26
Додаток Б……29

ВСТУП

Сьогодні неможливо уявити життя без використання електронно-обчислювальних пристроїв, адже вони дозволяють виконувати математичні та логічні операції з неймовірною швидкістю. Дисципліни електротехніка та схемотехніка займаються розробкою фізичних елементів ЕОМ та вивченням їх взаємодії. Але виконання складних операцій неможливе без програмування. Основні питання, що вирішує дисципліна архітектура комп'ютерів — організація елементів комп'ютера: оперативної пам'яті, процесора, зовнішніх пристроїв, створення систем команд, які буде виконувати процесор, організація пам'яті, створення програм для подальшого виконання їх процесором.

Для дослідження роботи та організації арифметико-логічного пристрою була створена модель двоадресної навчальної машини. Також розроблена та реалізована системакоманд (арифметичні операції, команди вводу-виводу тощо), що дозволяє отримувати чисельні дані від користувача, виконувати з ними різноманітні маніпуляції та обчислення, виводити результат користувачеві. Для зручності створено інтерфейс користувача, який дозволяє взаємодіяти з навчальною машиною: вводити та отримувати дані, запускати програми в звичайному та покроковому режимі, зберігати та відкривати створені програми.

СИСТЕМА КОМАНД

Використовуючи принципи фон Неймана, була сконструйована власна навчальна ЕОМ, розроблена та реалізована система команд, що дозволяє вводити та виводити дані, виконувати різноманітні арифметичні дії (додавання, віднімання, ділення, множення та пошук залишок від ділення), а також інші команди для перенесення даних з однієї ячійки в іншу тощо. Усі команди виконуються послідовно, за винятком безумовного та умовного переходів, які можуть змінити стандартний хід програми. Їх робота буде описана нижче. Кожна команда має два операнди (адреси), якими вона може керувати. Для арифметичних команд результат обчислення записується в рядку з адресою А1. Пам’ять навчальноїмашини складається з 512 рядків, що мають адреси від 1 до 512.

Нижче наведена таблиця реалізованих команд, їх кількість та короткий опис:

№	КОМ	ОПИС
0	ПЕР	Пересилання значення з А2 в А1
1	ПДЧ	Додавання дійсних чисел: А1 = А1 + А2
2	ВДЧ	Віднімання дійсних чисел: А1 = А1 - А2
3	МДЧ	Множення дійсних чисел: А1 = А1 × А2
4	ДДЧ	Ділення дійсних чисел: А1 = А1 ÷ А2
5	ВЕД	Введення масиву дійсних чисел у кількості А2, починаючи з адреси А1
6	ВЕЦ	Введення масиву цілих чисел у кількості А2, починаючи з адреси А1
7	БПР	Безумовний перехід з поточного рядка на рядок А2
8	ПДЦ	Переведення дійсного числа А1 в ціле А2

9	ПЦЧ	Додавання цілих чисел: A1 = A1 + A2
10	ВЧЦ	Віднімання цілих чисел: A1 = A1 - A2
11	МЦЧ	Множення цілих чисел: A1 = A1 × A2
12	ДЦЧ	Ділення цілих чисел: A1 = A1 ÷ A2
13	ВИД	Вивід масиву дійсних чисел у кількості A2, починаючи з адреси A1
14	ВИЦ	Вивід масиву цілих чисел у кількості A2, починаючи з адреси A1
15	УПЗ	Умовний перехід: якщо прапор Z дорівнює 0 — перехід на рядок A1, якщо прапор Z дорівнює 1 — перехід на рядок A2
16	УПЛ	Умовний перехід: якщо прапор ω дорівнює 0 або 2 — перехід на рядокA1, якщо прапор ω дорівнює 1 — перехід на рядок A2
17	ПЕЦ	Переведення цілого числа A1 у дійсне A2
18	МОД	Остача від ділення цілих чисел
19	ИТР	Зрушити A2 (задіяний зараз індекс ячійки масиву) в адресі A1 на A2 елементів
20	ОСТ	Завершення виконання програми

***Таблиця 1**. Система команд навчальної машини*

Команда пересилання

Команда пересилання має номер 0 та мнемонічний еквівалент “ПЕР”. Вона використовує два операнди: A1 та A2, де A1 є адресою призначення, а A2 є адресою джерела(тобто звідки треба брати значення, для пересилки в A1). За замовчуванням, усі рядки програми заповнені нулями, а в мнемонічному еквіваленті виглядають як “ПЕР 000 000”, що означає значення за адресою 0 записати в рядок за адресою нуль, тобто в нікуди. Фактично, такий рядок ніяк не впливає на виконання програми або інші рядки, і може слугувати як візуальний роздільник між значущими рядками.

Команда додавання дійсних чисел

Команда додавання дійсних чисел має код 1 і мнемонічний еквівалент “ПДЧ”. Для своєї роботи вона використовує два операнди: A1 та A2. A1 та A2 — адреси дійсних чиселзі знаком або без, які потрібно скласти, результат додавання записується в A1.

Команда віднімання дійсних чисел

Команда віднімання дійсних чисел має код 2 і мнемонічний еквівалент “ВДЧ”. Для своєї роботи вона використовує два операнди: A1 та A2. A1 та A2 — адреси дійсних чиселзі знаком або без, де A1 — це зменшуване, а A2 — від’ємник, результат віднімання записується в A1.

Команда множення дійсних чисел

Команда множення дійсних чисел має код 3 і мнемонічний еквівалент “МДЧ”. Для своєї роботи вона використовує два операнди: A1 та A2. A1 та A2 — адреси дійсних чиселзі знаком або без, де A1 та A2 — множники, результат множення записується в A1.

Команда ділення дійсних чисел

Команда ділення дійсних чисел має код 4 і мнемонічний еквівалент“ДДЧ”. Для своєїроботи вона використовує два операнди: А1 та А2. А1 та А2 — адреси дійсних чисел зі знаком або без, де А1 це ділене, А2 — дільник, результат ділення записується в А1.

Команда введення масиву дійсних чисел

Команда введення масиву дійсних чисел має код 5 і мнемонічний еквівалент “ВЕД”.Для своєї роботи вона використовує два операнди: А1 та А2. Де А1 — адреса рядка, з якоготреба починати записувати данні. Данні записуються послідовно. А2 — кількість чисел, які треба зчитати з клавіатури.

Команда введення масиву цілих чисел

Команда введення масиву цілих чисел має код 6 і мнемонічний еквівалент “ВЕЦ”. Для своєї роботи вона використовує два операнди: А1 та А2. Де А1 — адреса рядка, з якоготреба починати записувати данні. Данні записуються послідовно. А2 — кількість чисел, які треба зчитати з клавіатури. Команда аналогічна команді введення масиву дійсних чисел, окрім того, що зчитуються та записуються цілі числа.

Команда безумовного переходу

Команда безумовного переходу має код 7 і мнемонічний еквівалент “БПР”. Для своєї роботи вона використовує один операнд А2, який позначає номер рядка, який потрібно виконати наступним. Номер рядка не може перевищувати номер максимальної кількості рядків, тобто 512.

Команда переведення дійсного числа в ціле

Команда переведення дійсного числа в ціле має код 8 і мнемонічний еквівалент “ПДЦ”. Для своєї роботи вона використовує два операнди: А1 та А2. Де А2 — адреса рядка,куди потрібно записати перетворене ціле число, А1 — адреса рядка, звідки треба прочитатидійсне число, для подальшого його переведення в ціле.

Команда додавання цілих чисел

Команда додавання цілих чисел має код 9 і мнемонічний еквівалент “ПЦЧ”. Для своєї роботи вона використовує два операнди: A1 та A2. A1 та A2 — адреси цілих чисел зі знаком або без, які потрібно скласти, результат додавання записується в A1.

Команда віднімання цілих чисел

Команда віднімання дійсних чисел має код 10 і мнемонічний еквівалент “ВЦЧ”. Для своєї роботи вона використовує два операнди: A1 та A2. A1 та A2 — адреси цілих чисел зі знаком або без, де A1 це зменшуване, A2 — від’ємник, результат записується в A1.

Команда множення цілих чисел

Команда множення дійсних чисел має код 11 і мнемонічний еквівалент “МЦЧ”. Для своєї роботи вона використовує два операнди: A1 та A2. A1 та A2 — адреси цілих чисел зі знаком або без, де A1 та A2 множники, результат множення записується в A1.

Команда ділення цілих чисел

Команда ділення дійсних чисел має код 12 і мнемонічний еквівалент “ДЦЧ”. Для своєї роботи вона використовує два операнди: A1 та A2. A1 та A2 — адреси цілих чисел зі знаком або без, де A1 це ділене, а A2 — дільник, результат ділення записується в A1.

Команда виводу масиву дійсних чисел

Команда виводу масиву дійсних чисел має код 13 і мнемонічний еквівалент “ВИД”.Для своєї роботи вона використовує два операнди: A1 та A2. A1 — адреса рядка, з якого треба починати виводити данні. Данні виводяться послідовно, A2 - кількість чисел, які треба вивести на екран. Команда виводить масив дійсних чисел.

Команда виводу масиву цілих чисел

Команда виводу масиву дійсних чисел має код 14 і мнемонічний еквівалент “ВИЦ”.Для своєї роботи вона використовує два операнди: A1 та A2. Де A1 — адреса рядка, з якоготреба починати виводити данні. Данні виводяться послідовно, A2 — кількість чисел, які треба вивести на екран. Команда виводить масив цілих чисел.

Команда умовного переходу (залежна від прапора нульового результату)

Команда умовного переходу має код 15 і мнемонічний еквівалент “УПЗ”. Для своєїроботи вона використовує два операнди: A1 та A2. Кожен з операндів є адресою рядка, на який треба перейти, і який буде виконаний наступним. Перехід на один з двох рядків виконується за певними правилами. Для цього використовується значення прапору Z (нульового результату). Він оновлюється після виконання арифметичних операцій: додавання, віднімання, множення, ділення. Так, його значення буде 1, якщо результат операції дорівнює нулю. Якщо результат операції менше або більше за нуль, прапор буде встановлений в 0. Таким чином, в залежності від поточного значення прапору нульового результату, буде вирішено, на який рядок буде виконаний перехід. Якщо прапор дорівнює 1, програма переходить на рядок за адресою A1. У випадку, коли прапор дорівнює 0, наступним рядком стане рядок за адресою A2.

Команда умовного переходу (залежна від прапора ω)

Команда умовного переходу має код 16 і мнемонічний еквівалент “УПЛ”. Для своєїроботи вона використовує два операнди: A1 та A2. Кожен з операндів є адресою рядка, на який треба перейти, і який буде виконаний наступним. Перехід на один з двох рядків виконується за певними правилами. Для цього використовується значення прапору ω (Омега). Він оновлюється після виконання арифметичних операцій: додавання, віднімання,множення, ділення. Так, його значення буде 0, якщо результат операції дорівнює нулю. Якщо результат операції менше нуля, прапору буде встановлений у значення 1, якщо результат більше нуля, тоді прапору буде мати значення 2. Таким чином, в залежності від поточного значення прапору Омега, буде вирішено, на який рядок буде виконаний перехід. Якщо прапор омега дорівнює 0 або 2, програма переходить на рядок за адресою A1. У випадку, коли прапор омега дорівнює 1, наступним рядком стане рядок за адресою A2.

Команда переведення цілого числа в дійсне

Команда переведення цілого числа в дійсне має код 17 і мнемонічний еквівалент “ПЕЦ”. Для своєї роботи вона використовує два операнди A1 та A2. A1 — адреса рядка, куди потрібно записати перетворене дійсне число, A2 — адреса рядка, звідки треба прочитати ціле число, для подальшого його переведення в дійсне.

Команда остачі від ділення цілих чисел

Команда остачі від ділення дійсних чисел має код 18 і мнемонічний еквівалент “МОД”. Для своєї роботи вона використовує два операнди: A1 та A2. A1 та A2 — адреси цілих чисел зі знаком або без, де A1 — це ділене, а A2 — дільник, залишок, отриманий у результаті ділення, записується в адресу A1.

Команда ітерації масивом

Команда ітерації масивом має код 19 і мнемонічний еквівалент “ИТР”. Для своєї роботи вона використовує два операнди: A1 та A2. A1 — це адреса масиву, який потрібно зрушити, A2 — це кількість елементів, на яку потрібно зрушити індекс.

Команда зупинки виконання програми

Команда зупинки виконання програми має код 20 і мнемонічний еквівалент “ОСТ”.Для своєї роботи вона не потребує операндів. Результат виконання команди — зупинка виконання програми

РЕАЛІЗАЦІЯ НАВЧАЛЬНОЇ МАШИНИ

ІНСТРУМЕНТИ

Емулятор навчальної машини двоадресної була реалізована за допомогою мови програмування C# та з використанням фреймворку для побудови додатків з графічним користувацьким інтерфейсом під платформи операційних систем сімейства Windows, який має назву Microsoft WPF .Net Framework. Проект був виконаний у інтегрованому середовищі розробки Microsoft Visual Studio 2023.

ГРАФІЧНИЙ ІНТЕРФЕЙС КОРИСТУВАЧА

Для зручності користування навчальною машиною, був створений графічний інтерфейс користувача. Введення даних здійснюється за допомогою клавіатури, а управління емулятором за допомогою миші. Нижче наведений скріншот вікна, яке і є головним вікном, з яким взаємодіє користувач. Цифрами від 1 до 7 позначені елементи інтерфейсу, функціонування та призначення яких описано нижче.

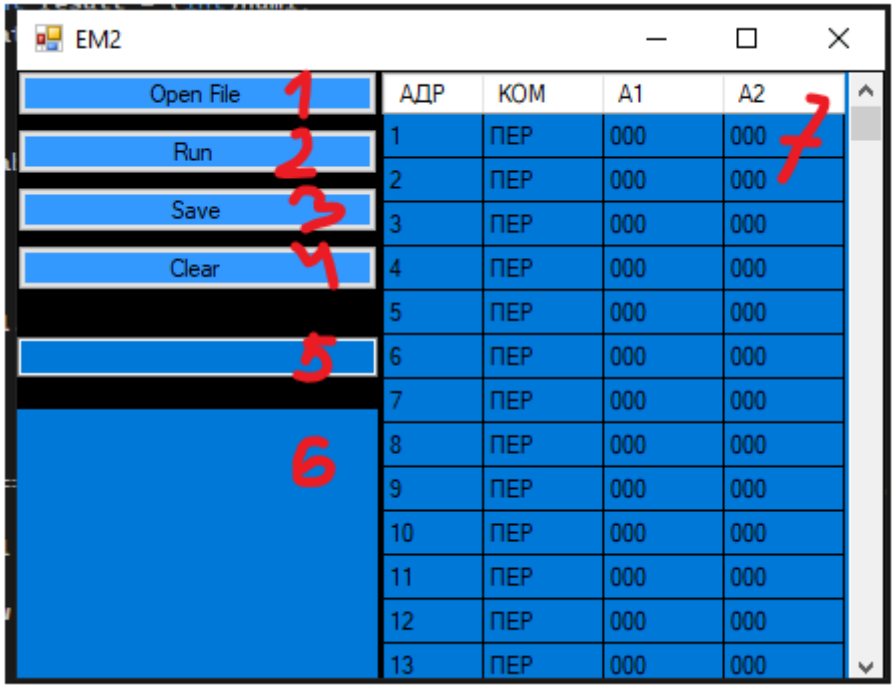

***Рисунок 1**. Графічний інтерфейс користувача*

Перелік та опис елементів інтерфейсу, позначених на рисунку 1:

1. Кнопка "Open" (Відкрити): Ця кнопка відкриває діалогове вікно, яке дозволяє користувачеві переглянути список файлів на комп'ютері. Вона дозволяє вибрати текстовий файл з розширенням .txt для подальшої обробки програмою. Це забезпечує зручний спосіб вибрати вхідні дані з файлу.

2. Кнопка "Run" (Виконати): Ця кнопка відповідає за запуск обробки всіх дій програми. Після натискання цієї кнопки програма виконує необхідні обчислення та дії згідно з введеними користувачем даними. Це дозволяє зручно та швидко запустити програму та отримати результати її роботи.

3. Кнопка "Clear" (Очистити): Ця кнопка служить для очищення даних, які відображаються в таблиці, мітках та текстовому полі. Після натискання цієї кнопки, дані, що були введені користувачем або відображені в інтерфейсі програми, будуть видалені, що дозволяє почати нову роботу з чистим інтерфейсом.

4. Кнопка "Save" (Зберегти): Ця кнопка використовується для збереження інформації з таблиці в текстовому форматі з певною структурою. Після натискання цієї кнопки програма зберігає дані з таблиці у вигляді текстового файлу з розширенням .txt. Це зручний спосіб зберегти результати обробки даних для подальшого використання.

5. TextBox: Це поле призначене для введення значень, які будуть оброблені командами програми. Користувач може ввести необхідні дані для обробки в цьому полі. Воно забезпечує зручний спосіб введення вхідних даних безпосередньо в інтерфейсі програми.

6. Label: Ця компонента інтерфейсу використовується для виведення повідомлень, попереджень та результатів роботи програми. Вона відображає текстову інформацію для сповіщення користувача про певні події або відображення результатів обробки даних. Мітка дозволяє зробити інтерфейс більш інформативним та зрозумілим для користувача.

7. Таблиця (DataGrid): Ця компонента інтерфейсу має 4 стовпці: АДР, КОМ, А1, А2. Вона служить як інтерфейс для взаємодії з обчислювальною машиною ЕМ2. Користувач може вводити дані у ці стовпці, змінювати їх і відображати результати обробки даних. Це забезпечує зручний та організований спосіб представлення даних та взаємодії з ними у програмі.

ПРОГРАМИ ДЛЯ НАВЧАЛЬНОЇ МАШИНИ

ЛІНІЙНІ ОБЧИСЛЕННЯ

Для тестування можливості реалізацій лінійних обчислень, складемо програму для подальшого її запуску за допомогою створеного емулятора навчальної машини.

17.Найти значение функции $y = \frac{4x^3 + 10z}{xz^2}$ при заданных значениях x и z.

значення “x” та “z” задає за допомогою операцій введення з клавіатури користувач.

Нижче наведений код відповідної програми у вигляді таблиці:

АДР	КОМ	А1	А2	ОПИС
1	ВЕД	15	2	Ввід користувачем значень “x” та “z” у 15 та 16 адреси
2	ПЕР	17	15	Пересилання “x” до 17 адреси, щоб пізніше не втратити значення
3	ПЕР	18	16	Пересилання “z” до 18 адреси, щоб пізніше не втратити значення
4	МДЧ	17	15	Множення “x” на “x” (піднесення до квадрату)
5	МДЧ	17	15	Множення “x^2” на “x” (піднесення до кубу)
6	МДЧ	14	17	Множення “x^3” на 4, заздалегідь переслану до 14 адреси
7	МДЧ	18	16	Множення “z” на “z” (піднесення до квадрату)
8	МДЧ	18	15	Множення “z^2” на “x” (обрахунок дільника)
9	МДЧ	19	16	Множення “z” на 10, заздалегідь переслану до 19 адреси
10	ПДЧ	14	19	Додавання $4x^3$ та 10z (обрахунок діленого)
11	ДДЧ	14	18	Фінальне ділення заздалегідь обрахованих значень

12	ВИД	14	1	Вивід обрахованого результату функції
13	ОСТ	000	000	Зупинка виконання програми
14	ПЕР	000	4	Пересилання до 14 адресу 4
…	…	…	…	…
19	ПЕР	000	10	Пересилання до 19 адресу 10

***Таблиця 2**. Код програми обчислення лінійного виразу*

ДЕМОНСТРАЦІЯ ВИКОНАННЯ ПРОГРАМИ

Ввівши значення “x” = 4.573 та “z” = 2.521 отримали результат 14.02925…

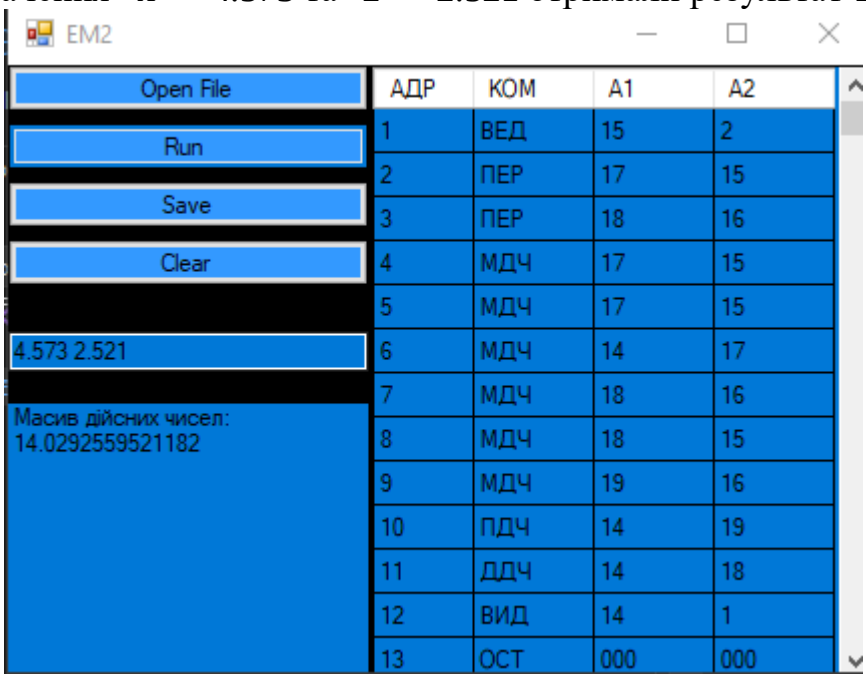

***Рисунок 4**. Обчислення лінійного виразу за допомогою емулятора навчальної машини*

Перевіримо отриманий результат за допомогою онлайн-калькулятора Wolfram Alpha:

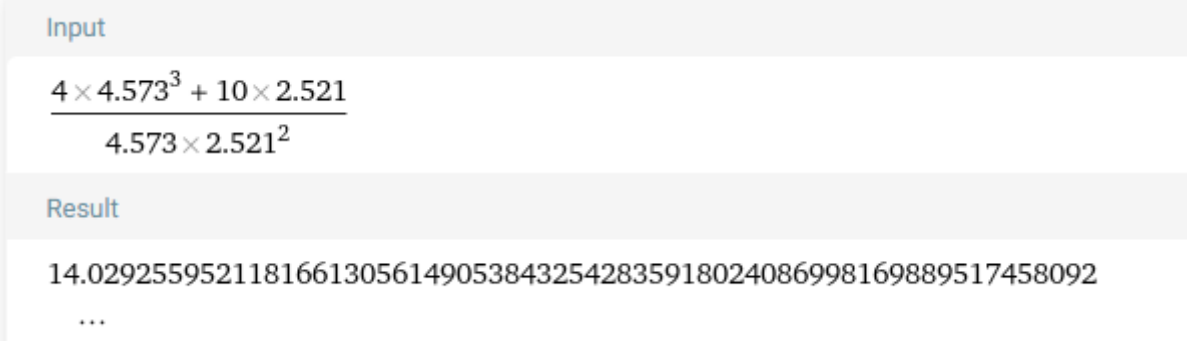

***Рисунок 5.** Обчислення лінійного виразу за допомогою калькулятора Wolfram Alpha*

ОБЧИСЛЕННЯ ЗА ДОПОМОГОЮ ЦИКЛІВ

Для тестування можливості реалізацій циклічних обчислень, складемо відповідну програму для подальшого її запуску за допомогою створеного емулятора навчальної машини.

2. Вычислить $y = \begin{cases} \prod_{i=1}^{n} \frac{i^3}{a-b} & \text{при } a \neq b \\ \frac{a-b}{a+b} & \text{в противном случае} \end{cases}$

Нижче наведений код відповідної програми у вигляді таблиці:

АДР	КОМ	А1	А2	ОПИС
1	ВЕД	40	3	Ввід користувачем значень “a”, “b” та “i” у 40, 41 та 42 адреси
2	ПЕР	50	40	Пересилка “a” до 50 адреси, щоб пізніше не втратити значення
3	ПЕР	51	41	Пересилка “b” до 51 адреси, щоб пізніше не втратити значення
4	ВДЧ	40	41	Перевірка, чи дорівнює a - b нулю(формування прапору нульового результату)
5	УПЗ	6	8	Якщо прапор зараз 0 (результат операції не дорівнював 0), то перейти до 8 адреси, інакше – до 6
6	БПР	000	20	Безумовний перехід до 20 адреси для подальшого обрахування результату
7	ОСТ	000	000	Зупинка виконання програми

8	ВДЧ	42	55	Перевіряємо, чи не більше ітератор(зберігається у 55 адресі), аніж потрібна користувачу кількість ітерацій (зберігається у 42 адресі). Формуємо прапор ω: якщо результатвіднімання більше або дорівнює 0, то ω = 2 (0), інакше ω = 1
9	УПЛ	10	18	Якщо обрахований прапор ω дорівнює 0 або 2, то переходимо до 10 адреси, інакше (дорівнює 1) до 18
10	ПДЧ	42	55	Повертаємо необхідну користувачукількість ітерацій до початкового значення (до формування ω)
11	ПЕР	57	55	Пересилання значення ітератору до
12	МДЧ	57	55	Множення “i” на “i” (піднесення до 57 адреси, щоб не втратити значення квадрату)
13	МДЧ	57	55	Множення “i^2” на “i” (піднесення до кубу)
14	ДДЧ	57	40	Ділення i^3 на обраховану раніше різницю a - b
15	МДЧ	512	57	Множимо результат на щойно обраховане значення
16	ПДЧ	55	56	Збільшуємо ітератор на одиницю
17	БПР	000	8	Безумовний перехід до 8 адреси для продовження (або завершення) циклу
18	ВИД	512	1	Виводимо результат обрахунку на екран
19	ОСТ	000	000	Зупинка виконання програми
20	ПДЧ	50	51	Оскільки a - b вже було обраховано у 4 адресі (результат у 40), залишається обрахувати лише суму
21	ДДЧ	40	50	Ділимо різницю a - b на тільки що обраховану суму a + b

22	ВИД	40	1	Виводимо результат обрахунку на екран
23	БПР	000	7	Переходимо до 7 адреси для завершення програми
…	…	…	….	…
55	ПЕР	000	1	Пересилання до 55 адреси 1
56	ПЕР	000	1	Пересилання до 56 адреси 1
…	…	…	…	…
511	ПЕР	000	0	Пересилання до 511 адреси 0, для обрахунку суми
512	ПЕР	000	1	Пересилання до 512 адреси 1, для обрахунку добутку

***Таблиця 3**. Код програми обчислення виразу за допомогою циклів*

ДЕМОНСТРАЦІЯ ВИКОНАННЯ ПРОГРАМИ

Ввівши значення “a” = 6.43 та “b” = 2.894, “i” = 5 отримали результат

1039739516,81

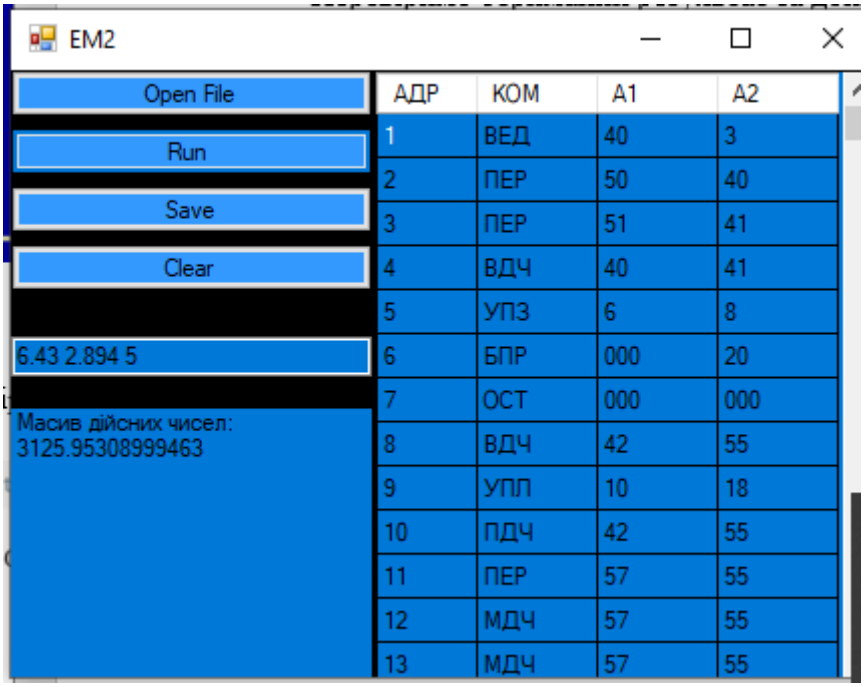

***Рисунок 6**. Обчислення циклічного виразу за допомогою емулятора навчальної машини*

Перевіримо отриманий результат за допомогою онлайн-калькулятора Wolfram Alpha:

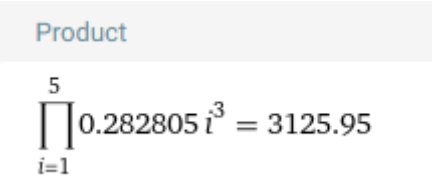

***Рисунок 7.** Обчислення циклічного виразу за допомогою калькулятора Wolfram Alpha*

Ввівши значення “a” = 2 та “b” = 2, “i” = 5 отримали результат 0. Оскільки $a = b$, програма обрахувала значення (a-b)/(a+b) а оскільки “a” та “b” мають однакові значення, при відніманні у чисельнику дробу завжди з’являється нуль, тому й відповідь завжди буде 0.

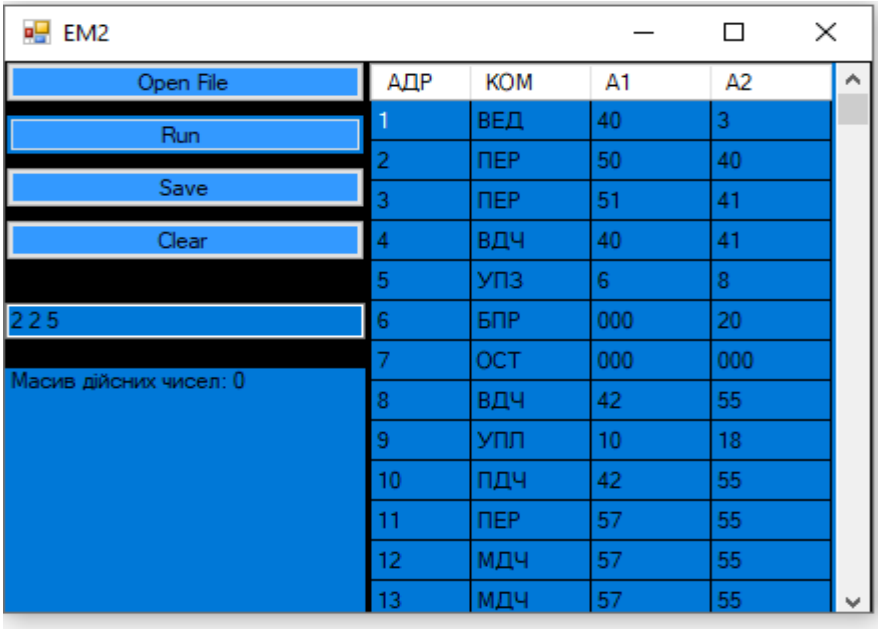

***Рисунок 8**. Обчислення циклічного виразу за допомогою емулятора навчальної машини*

Як можна переконатись, порівнявши результати обчислень на зображеннях, емулятор рахуєдосить точно і абсолютно коректно розуміє програму, написану для тестування.

ОБЧИСЛЕННЯ З ВИКОРИСТАННЯМ ЦИКЛІВ (МАСИВИ)

Для тестування можливості реалізацій обчислень масивів з використанням циклів, складемо відповідну програму для подальшого її запуску за допомогою створеного емулятора навчальної машини.

Завдання №11: Скласти програму для підрахунку в цілочисловому масиві кількості нульових елементів, що стоять після елемента із заданим користувачем номером.

Нижче наведений код відповідної програми у вигляді таблиці:

АДР	КОМ	А1	А2	ОПИС
1	ВЕЦ	100	9	Заповнення користувачем масиву з 8 и адресу числа з якого рахування нулів елементів, починаючи з 100 адресу
2	ВЦЧ	108	30	Перевірка, чи досяг ітератор (у 30 адресі) початкового значення (яке ввів користувач до 108 адресу) дляпочатку розрахунків. Формування прапору нульового результату
3	УПЗ	4	6	Умовний перехід: якщо ітератор ще не досяг початкового індексу (прапор дорівнює 0) перехід до 6 адресу, інакше (прапор 1) до 4 адресу
4	ПЦЧ	108	30	Повертаємо початковий індекс до його стартового значення (яке змінилося після виконання коду за 2 адресою)
5	БПР	000	9	Безумовний перехід до 10 адресу
6	ПЦЧ	108	30	Повертаємо початковий індекс до його стартового значення (яке змінилося після виконання коду за 2 адресою)
7	ПЦЧ	30	31	Ітератор ще не досяг початкового індексу, збільшуємо його на один
8	БЕЗ	000	2	Безумовний перехід до 2 адресу
9	ВЧЦ	32	30	Перевіряємо, чи не більше ітератор (зберігається у 30 адресі), аніжкількість елементів масиву(зберігається у 32 адресі)

10	УПЛ	11	20	Якщо значення прапору ω дорівнює0 (або 2) то продовжуємо цикл і переходимо до 11 адресу, інакше (ω дорівнює 1) до 20 адресу
11	ПЦЧ	32	30	Повертаємо початкову кількість елементів масиву до її стартового значення (яке змінилося після виконання коду за 9 адресою)
12	ПЦЧ	100	34	Додаю до поточного елементу масиву (якій змінюється на 1 після виконання коду та 13 адресою) нуль. Формую прапор нульового результату: якщо елемент масиву бувнулем, то прапор буде дорівнювати 1, інакше 0
13	ИТР	12	1	Зрушуємо адрес (А1) необхідної ячійки масиву у 13 адресі на один
14	УПЗ	17	15	Умовний перехід: якщо прапор нульового результату дорівнює 0 до15 адреси, інакше (прапор дорівнює 1) до 17 адреси
15	ПЦЧ	30	31	Збільшую ітератор на одиницю
16	БПР	000	10	Безумовний перехід до 9 адреси
17	ПЦЧ	30	31	Збільшую ітератор на одиницю
18	ПЦЧ	33	31	Збільшую кількість нулів у масиві на одиницю (спочатку за 33 адресою зберігається 0)
19	БПР	000	10	Безумовний перехід до 9 адреси
20	ВИЦ	33	1	Вивід кількості нулів на екран
21	ОСТ	000	000	Зупинка виконання програми
…	…	…	…	…
30	ПЕР	000	1	Пересилання до 30 адреси 1
31	ПЕР	000	1	Пересилання до 31 адреси 1
32	ПЕР	000	5	Пересилання до 32 адреси 5
34	ПЕР	000	0	Пересилання до 34 адреси 0

***Таблиця 4**. Код програми обчислення масивів за допомогою циклів*

ДЕМОНСТРАЦІЯ ВИКОНАННЯ ПРОГРАМИ

Після заповнення масиву цифрами 1, 0, 3, 0, 5 , 6 , 0 і вводу 2 (елемент, з якого починаєтьсяпідрахунок кількості нулів) отримали очікуваний результат 3.

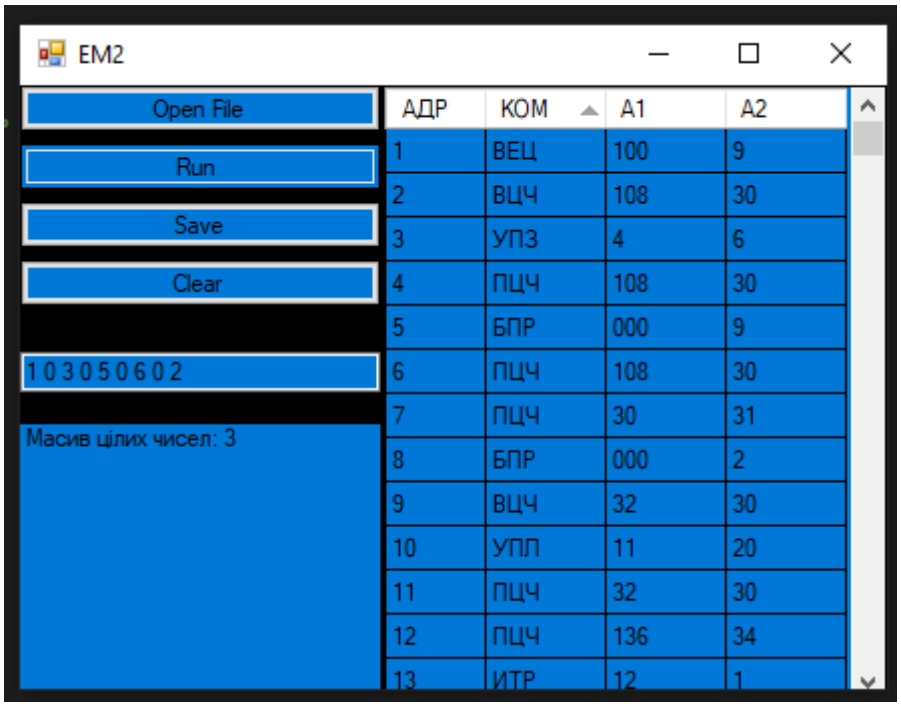

***Рисунок 9**. Обчислення масивів із використанням циклів за допомогою емулятора навчальної машини*

Як можна переконатись, емулятор цілі числа рахує точно і абсолютно коректно розумієпрограму, написану для тестування.

ВИСНОВКИ

У даній курсовій роботі був реалізований емулятор навчальної двоадресної машини із сучасним користувацьким інтерфейсом. Була розроблена та реалізована система команд, за допомогою яких, користувач може вводити/виводити дані та обчислювати різні лінійні вирази, створювати програми з використанням циклів та масивів. Зберігати та відкривати існуючі файли з програмами для навчальної машини. Також був розроблений режим налагодження, за допомогою якого, користувач має змогу самостійно запускати рядки програми крок за кроком, при чому спостерігаючи стан усіх необхідних регістрів у відповідному вікні. У разі виникнення помилок чи непередбачуваних ситуацій, була створена система оповіщення про це користувача - у відповідному вікні виводиться інформація про саму помилку та рядок, на якому вона трапилася.

Для тестування емулятора навчальної машини, були написані програми відповідно до завдань до варіанту, які тестували обчислення емулятором лінійних програм, програм із циклами, та можливість використовувати обчислення на масивах. Для перевірки коректності відповіді використовувалася система Wolfram Alpha, за допомогою якої, було встановлено, що результати обчислення програм збігаються. Ґрунтуючись на цих дослідженнях, та невеликих тестуваннях під час розробки емулятора навчальної машини, був зроблений висновок, що емулятор працює правильно і дозволяє створювати повноцінні програми для проведення обчислень із цілими та дійсними числами.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. *Аванесян Г.Р., Лёвшин В.П.* Интегральные микросхемы ТТЛ, ТТЛШ: Справочник. М.:Машиностроение, 1993.

2. *Атовмян И.О.* Архитектура вычислительных систем. М.: МИФИ, 2002.

3. *Бродин В.Б., Шагурин И. И.* Микропроцессор i486. Архитектура, программирование,интерфейс. М.: ДИАЛОГ-МИФИ, 1993.

4. *Гибсон Г., Лю Ю-Чжен.* Микропроцессоры семейства 8086/8088. М.: Радио и связь,1987.

5. *Гордеев А.В.* Операционные системы: Учебник для вузов. 2-е изд. СПб.: Питер, 2007.

6. *Гуров В.В.* Синтез комбинационных схем в примерах. М.: МИФИ, 2001.

7. *Гуров В.В., Ленский О.Д., Соловьев Г.Н., Чуканов В.О.* Архитектура, структура и организация вычислительного процесса в ЭВМ типа IBM PC / Подред. Г.Н. Соловьева. М.:МИФИ, 2002.

8. *Гуров В.В., Чуканов В.О.* Электронная книга. Архитектура и организация ЭВМ. М.:ИНТУИТ. Национальный открытый университет, 2005.

9. *Дэвид Харрис, Сара Харрис.* Цифровая схемотехника и архитектура компьютера. 2-еизд.: перевод командой компаний и университетов России, Украины, США и Великобритании, Morgan Kaufman, 2013.

10. *Жмакин А.П.* Архитектура ЭВМ. 2-е изд. СПб.: БХВ-Петербург, 2010.

11. *Каган Б.М.* Электронные вычислительные машины и системы. М.: Энергоатомиздат,1991.

12. *Казаринов Ю.М., Номоконов В.Н., Подклетнов Г.С.* и др. Микропроцессорный комплект К1810: Структура, программирование, применение / Подред. Ю.М. Казаринова.М.: Высшая школа, 1990.

13. *Киселев А.В., Корнеев В.В.* Современные микропроцессоры. М.: Но-лидж, 1998.

14. *Майоров С.А., Кириллов В.В., Приблуда А.А.* Введение в микроЭВМ. Ленинград:Машиностроение, 1988.

Додаток А

Код фалу EM2.Designer.cs, відповідаючий за інтерфейс:

```
using System.Data.Common;
using System.Windows.Forms;
namespace k1
{
  partial class EM2
  {
    /// <summary>
    /// Обязательная переменная конструктора./// </summary>
    private System.ComponentModel.IContainer components = null;
    /// <summary>
    /// Освободить все используемые ресурсы./// </summary>
    /// <param name="disposing">истинно, если управляемый ресурс должен быть удален; иначе ложно.</param>
    protected override void Dispose(bool disposing)
    {
      if (disposing && (components != null))
      {
        components.Dispose();
      }
      base.Dispose(disposing);
    }
    #region Код, автоматически созданный конструктором форм Windows
    /// <summary>
    /// Требуемый метод для поддержки конструктора — не изменяйте
    /// содержимое этого метода с помощью редактора кода./// </summary>
    private void InitializeComponent()
    {
      this.button1 = new System.Windows.Forms.Button();
      this.button2 = new System.Windows.Forms.Button();
      this.button3 = new System.Windows.Forms.Button();
      this.textBox1 = new System.Windows.Forms.TextBox();
      this.dataGridView1 = new System.Windows.Forms.DataGridView();
      this.Column1 = new System.Windows.Forms.DataGridViewTextBoxColumn();
      this.Column2 = new System.Windows.Forms.DataGridViewTextBoxColumn();
      this.Column3 = new System.Windows.Forms.DataGridViewTextBoxColumn();
      this.Column4 = new System.Windows.Forms.DataGridViewTextBoxColumn();
      this.label1 = new System.Windows.Forms.Label();
      this.button4 = new System.Windows.Forms.Button();
      ((System.ComponentModel.ISupportInitialize)(this.dataGridView1)).BeginInit();
      this.SuspendLayout();
      //
      // button1
      //
      this.button1.BackColor = System.Drawing.SystemColors.MenuHighlight;
      this.button1.Location = new System.Drawing.Point(0, 0);
      this.button1.Name = "button1";
      this.button1.Size = new System.Drawing.Size(180, 23);
      this.button1.TabIndex = 0;
      this.button1.Text = "Open File";
      this.button1.UseVisualStyleBackColor = false;
      //
      // button2
      //
      this.button2.BackColor = System.Drawing.SystemColors.MenuHighlight;
      this.button2.Location = new System.Drawing.Point(0, 29);
      this.button2.Name = "button2";
      this.button2.Size = new System.Drawing.Size(180, 23);
      this.button2.TabIndex = 1;
      this.button2.Text = "Run ";
      this.button2.UseVisualStyleBackColor = false;
      //
      // button3
      //
      this.button3.BackColor = System.Drawing.SystemColors.MenuHighlight;
      this.button3.Location = new System.Drawing.Point(0, 58);
```

```
this.button3.Name = "button3";
this.button3.Size = new System.Drawing.Size(180, 23);
this.button3.TabIndex = 2;
this.button3.Text = "Save";
this.button3.UseVisualStyleBackColor = false;
//
// textBox1
//
this.textBox1.Anchor = ((System.Windows.Forms.AnchorStyles)((((System.Windows.Forms.AnchorStyles.Top |
System.Windows.Forms.AnchorStyles.Bottom)
| System.Windows.Forms.AnchorStyles.Left)
| System.Windows.Forms.AnchorStyles.Right)));
this.textBox1.BackColor = System.Drawing.SystemColors.Highlight;
this.textBox1.Location = new System.Drawing.Point(0, 133);
this.textBox1.Name = "textBox1";
this.textBox1.Size = new System.Drawing.Size(180, 20);
this.textBox1.TabIndex = 3;
//
// dataGridView1
//
this.dataGridView1.BackgroundColor = System.Drawing.SystemColors.Highlight;
this.dataGridView1.ColumnHeadersHeightSizeMode =
System.Windows.Forms.DataGridViewColumnHeadersHeightSizeMode.AutoSize;
this.dataGridView1.Columns.AddRange(new System.Windows.Forms.DataGridViewColumn[] {
this.Column1,
this.Column2,
this.Column3,
this.Column4});
this.dataGridView1.GridColor = System.Drawing.SystemColors.ActiveCaptionText;
this.dataGridView1.Location = new System.Drawing.Point(181, 0);
this.dataGridView1.Name = "dataGridView1";
this.dataGridView1.RowHeadersVisible = false;
this.dataGridView1.Size = new System.Drawing.Size(252, 307);
this.dataGridView1.TabIndex = 5;
this.dataGridView1.CellContentClick += new
System.Windows.Forms.DataGridViewCellEventHandler(this.dataGridView1_CellContentClick);
//
// Column1
//
this.Column1.HeaderText = "АДР";
this.Column1.Name = "Column1";
this.Column1.ReadOnly = true;
this.Column1.Width = 50;
//
// Column2
//
this.Column2.HeaderText = "КОМ";
this.Column2.Name = "Column2";
this.Column2.Width = 60;
//
// Column3
//
this.Column3.HeaderText = "A1";
this.Column3.Name = "Column3";
this.Column3.Width = 60;
//
// Column4
//
this.Column4.HeaderText = "A2";
this.Column4.Name = "Column4";
this.Column4.Width = 60;
//
// label1
//
this.label1.Anchor = ((System.Windows.Forms.AnchorStyles)((((System.Windows.Forms.AnchorStyles.Top |
System.Windows.Forms.AnchorStyles.Bottom)
```

```
            | System.Windows.Forms.AnchorStyles.Left)
            | System.Windows.Forms.AnchorStyles.Right)));
            this.label1.BackColor = System.Drawing.SystemColors.Highlight;
            this.label1.Location = new System.Drawing.Point(0, 169);
            this.label1.Name = "label1";
            this.label1.Size = new System.Drawing.Size(180, 138);
            this.label1.TabIndex = 4;
            //
            // button4
            //
            this.button4.BackColor = System.Drawing.SystemColors.MenuHighlight;
            this.button4.Location = new System.Drawing.Point(0, 87);
            this.button4.Name = "button4";
            this.button4.Size = new System.Drawing.Size(180, 23);
            this.button4.TabIndex = 6;
            this.button4.Text = "Clear";
            this.button4.UseVisualStyleBackColor = false;
            this.button4.Click += new System.EventHandler(this.button4_Click);
            //
            // EM2
            //
            this.AutoScaleDimensions = new System.Drawing.SizeF(6F, 13F);
            this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;
            this.BackColor = System.Drawing.SystemColors.InfoText;
            this.ClientSize = new System.Drawing.Size(433, 303);
            this.Controls.Add(this.button4);
            this.Controls.Add(this.dataGridView1);
            this.Controls.Add(this.label1);
            this.Controls.Add(this.textBox1);
            this.Controls.Add(this.button3);
            this.Controls.Add(this.button2);
            this.Controls.Add(this.button1);
            this.Name = "EM2";
            this.Text = "EM2";
    ((System.ComponentModel.ISupportInitialize)(this.dataGridView1)).EndInit();
            this.ResumeLayout(false);
            this.PerformLayout();

        }
        #endregion
        private System.Windows.Forms.Button button1;
        private System.Windows.Forms.Button button2;
        private System.Windows.Forms.Button button3;
        private System.Windows.Forms.TextBox textBox1;
        private System.Windows.Forms.DataGridView dataGridView1;
        private DataGridViewTextBoxColumn Column1;
        private DataGridViewTextBoxColumn Column2;
        private DataGridViewTextBoxColumn Column3;
        private DataGridViewTextBoxColumn Column4;
        private Label label1;
        private Button button4;
    }
}
```

Додаток Б

Код файлу EM2.cs реализуючий всю логіку програми:

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
```

```
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.IO;
using System.Reflection.Emit;
using static System.Windows.Forms.VisualStyles.VisualStyleElement;
using System.Security.Cryptography;

namespace k1
{
    public partial class EM2 : Form
    {
        public EM2()
        {
            InitializeComponent();
            button1.Click += button1_Click;
            button2.Click += button2_Click;
            button3.Click += button3_Click;
            button4.Click += button4_Click;
        }
        private void button1_Click(object sender, EventArgs e)
        {
            OpenFileDialog openFileDialog = new OpenFileDialog();
            openFileDialog.Filter = "Text Files (*.txt)|*.txt";
            openFileDialog.InitialDirectory = Environment.GetFolderPath(Environment.SpecialFolder.MyComputer);

            if (openFileDialog.ShowDialog() == DialogResult.OK)
            {
                string filePath = openFileDialog.FileName;
                string[] lines = File.ReadAllLines(filePath);

                foreach (string line in lines)
                {
                    string[] parts = line.Split(':');

                    if (parts.Length == 2)
                    {
                        string adr = parts[0].Trim();
                        string[] values = parts[1].Trim().Split(' ');

                        if (values.Length == 3)
                        {
                            string kom = values[0].Trim();
                            string a1 = values[1].Trim();
                            string a2 = values[2].Trim();

                            DataGridViewRow rowToUpdate = null;
                            foreach (DataGridViewRow row in dataGridView1.Rows)
                            {
                                if (row.Cells[0].Value != null && row.Cells[0].Value.ToString() == adr)
                                {
                                    rowToUpdate = row;
                                    break;
                                }
                            }

                            if (rowToUpdate != null)
                            {
                                rowToUpdate.Cells[1].Value = kom;
                                rowToUpdate.Cells[2].Value = a1;
                                rowToUpdate.Cells[3].Value = a2;
                            }
                        }
```

```
                }
            }
        }
    }
    private void button2_Click(object sender, EventArgs e)
    {


        int currentRow = 0;
        int zFlag = 0;
        int omegaFlag = 0;
        while (currentRow < dataGridView1.Rows.Count)
        {
            DataGridViewRow row = dataGridView1.Rows[currentRow];

            if (row.Cells[0].Value != null && row.Cells[1].Value != null && row.Cells[2].Value != null &&
row.Cells[3].Value != null)
            {
                string adr = row.Cells[0].Value.ToString();
                string command = row.Cells[1].Value.ToString();
                string a1 = row.Cells[2].Value.ToString();
                string a2 = row.Cells[3].Value.ToString();


                if (command == "ПЕР") //Пересилання значення з A2 в A1
                {
                    float a1Value = float.Parse(a1);
                    float a2Value = float.Parse(a2);

                    int a1Index = (int)a1Value - 1;
                    int a2Index = (int)a2Value - 1;

                    if (a1Index >= 0 && a1Index < dataGridView1.Rows.Count && a2Index >= 0 && a2Index <
dataGridView1.Rows.Count)
                    {
                        string valueToTransferA2 = dataGridView1.Rows[a2Index].Cells[3].Value.ToString();

                        dataGridView1.Rows[a1Index].Cells[3].Value = valueToTransferA2;
                    }
                }

                else if (command == "ПДЧ")//Додавання дійсних чисел: A1 = A1 + A2

                {
                    int a1Index = int.Parse(a1) - 1;
                    int a2Index = int.Parse(a2) - 1;

                    if (a1Index >= 0 && a1Index < dataGridView1.Rows.Count && a2Index >= 0 && a2Index <
dataGridView1.Rows.Count)
                    {
                        string valueA1 = dataGridView1.Rows[a1Index].Cells[3].Value.ToString();
                        string valueA2 = dataGridView1.Rows[a2Index].Cells[3].Value.ToString();

                        double num1 = double.Parse(valueA1);
                        double num2 = double.Parse(valueA2);
                        double sum = num1 + num2;
                        if (sum == 0)
                        {
                            omegaFlag = 0;
                            zFlag = 1;
                        }
                        else if (sum < 0)
                        {
```

```
                omegaFlag = 1;
                zFlag = 0;
            }
            else
            {
                omegaFlag = 2;
                zFlag = 0;
            }
            dataGridView1.Rows[a1Index].Cells[3].Value = sum.ToString();
        }
    }

    else if (command == "ВДЧ")//Віднімання дійсних чисел: A1 = A1 - A2
    {
        int a1Index = int.Parse(a1) - 1;
        int a2Index = int.Parse(a2) - 1;

        if (a1Index >= 0 && a1Index < dataGridView1.Rows.Count && a2Index >= 0 && a2Index <
dataGridView1.Rows.Count)
        {
            string valueA1 = dataGridView1.Rows[a1Index].Cells[3].Value.ToString();
            string valueA2 = dataGridView1.Rows[a2Index].Cells[3].Value.ToString();

            double num1 = double.Parse(valueA1);
            double num2 = double.Parse(valueA2);
            double raz = num1 - num2;
            if (raz == 0)
            {
                omegaFlag = 0;
                zFlag = 1;
            }
            else if (raz < 0)
            {
                omegaFlag = 1;
                zFlag = 0;
            }
            else
            {
                omegaFlag = 2;
                zFlag = 0;
            }
            dataGridView1.Rows[a1Index].Cells[3].Value = raz.ToString();
        }
    }

    else if (command == "МДЧ")//Множення дійсних чисел: A1 = A1 × A2
    {
        int a1Index = int.Parse(a1) - 1;
        int a2Index = int.Parse(a2) - 1;

        if (a1Index >= 0 && a1Index < dataGridView1.Rows.Count && a2Index >= 0 && a2Index <
dataGridView1.Rows.Count)
        {
            string valueA1 = dataGridView1.Rows[a1Index].Cells[3].Value.ToString();
            string valueA2 = dataGridView1.Rows[a2Index].Cells[3].Value.ToString();

            double num1 = double.Parse(valueA1);
            double num2 = double.Parse(valueA2);
            double mno = num1 * num2;
            if (mno == 0)
            {
                omegaFlag = 0;
                zFlag = 1;
```

```
                }
                else if (mno < 0)
                {
                    omegaFlag = 1;
                    zFlag = 0;
                }
                else
                {
                    omegaFlag = 2;
                    zFlag = 0;
                }
                dataGridView1.Rows[a1Index].Cells[3].Value = mno.ToString();
            }
        }


        else if (command == "ДДЧ")//Ділення дійсних чисел: A1 = A1 ÷ A2
        {
            int a1Index = int.Parse(a1) - 1;
            int a2Index = int.Parse(a2) - 1;

            if (a1Index >= 0 && a1Index < dataGridView1.Rows.Count && a2Index >= 0 && a2Index <
dataGridView1.Rows.Count)
                {
                string valueA1 = dataGridView1.Rows[a1Index].Cells[3].Value.ToString();
                string valueA2 = dataGridView1.Rows[a2Index].Cells[3].Value.ToString();
                double num1 = double.Parse(valueA1);
                double num2 = double.Parse(valueA2);
                 if (num2 != 0)
                 {
                     double del = num1 / num2;
                    if (del == 0)
                    {
                        omegaFlag = 0;
                        zFlag = 1;
                    }
                    else if (del < 0)
                    {
                        omegaFlag = 1;
                        zFlag = 0;
                    }
                    else
                    {
                        omegaFlag = 2;
                        zFlag = 0;
                    }
                    dataGridView1.Rows[a1Index].Cells[3].Value = del.ToString();
                 }
                 else
                 {
                     label1.Text = "ДДЧ: Деление на 0.";
                     return;
                 }
            }
        }

        else if (command == "ВЕД")//Введення масиву дійсних чисел у кількості A2, починаючи з адреси
A1
        {
            int startingRow = int.Parse(a1) - 1;
            int count = int.Parse(a2);

            string inputValues = textBox1.Text.Trim();
```

```
            string[] numbers = inputValues.Split(' ');

            if (numbers.Length == count)
            {
                for (int i = 0; i < count; i++)
                {
                    double value = double.Parse(numbers[i]);
                    dataGridView1.Rows[startingRow + i].Cells[3].Value = value;
                }
            }
            else
            {
                label1.Text = "ВЕД: Количество введенных чисел не соответствует значению A2.";
                return;
            }
        }

        else if (command == "ВЕЦ")//Введення масиву цілих чисел у кількості A2, починаючи з адреси
A1
        {
            int startingRow = int.Parse(a1) - 1;
            int count = int.Parse(a2);

            string inputValues = textBox1.Text.Trim();
            string[] numbers = inputValues.Split(' ');

            if (numbers.Length == count)
            {
                for (int i = 0; i < count; i++)
                {
                    if (int.TryParse(numbers[i], out int value))
                    {
                        dataGridView1.Rows[startingRow + i].Cells[3].Value = value;
                    }
                    else
                    {
                        label1.Text = "ВЕЦ: Введены некорректные значения чисел.";
                    }
                }
            }
            else
            {
                label1.Text = "ВЕЦ: Количество введенных чисел не соответствует значению A2.";
            }
        }

        else if (command == "ПЦЧ")//Додавання цілих чисел: A1 = A1 + A2
        {
            int a1Index = int.Parse(a1) - 1;
            int a2Index = int.Parse(a2) - 1;

            if (a1Index >= 0 && a1Index < dataGridView1.Rows.Count && a2Index >= 0 && a2Index <
dataGridView1.Rows.Count)
            {
                string valueA1 = dataGridView1.Rows[a1Index].Cells[3].Value.ToString();
                string valueA2 = dataGridView1.Rows[a2Index].Cells[3].Value.ToString();

                if (int.TryParse(valueA1, out int num1) && int.TryParse(valueA2, out int num2))
                {
                    int sum = num1 + num2;
                    if (sum == 0)
                    {
                        omegaFlag = 0;
```

```
                zFlag = 1;
            }
            else if (sum < 0)
            {
                omegaFlag = 1;
                zFlag = 0;
            }
            else
            {
                omegaFlag = 2;
                zFlag = 0;
            }
            dataGridView1.Rows[a1Index].Cells[3].Value = sum.ToString();
        }
        else
        {
            label1.Text = "ПЦЧ: Введены некорректные значения чисел.";
        }
    }
}

else if (command == "ВЦЧ")//Віднімання цілих чисел: A1 = A1 - A2
{
    int a1Index = int.Parse(a1) - 1;
    int a2Index = int.Parse(a2) - 1;

    if (a1Index >= 0 && a1Index < dataGridView1.Rows.Count && a2Index >= 0 && a2Index <
dataGridView1.Rows.Count)
    {
        string valueA1 = dataGridView1.Rows[a1Index].Cells[3].Value.ToString();
        string valueA2 = dataGridView1.Rows[a2Index].Cells[3].Value.ToString();

        if (int.TryParse(valueA1, out int num1) && int.TryParse(valueA2, out int num2))
        {
            int raz = num1 - num2;
            if (raz == 0)
            {
                omegaFlag = 0;
                zFlag = 1;
            }
            else if (raz < 0)
            {
                omegaFlag = 1;
                zFlag = 0;
            }
            else
            {
                omegaFlag = 2;
                zFlag = 0;
            }
            dataGridView1.Rows[a1Index].Cells[3].Value = raz.ToString();
        }
        else
        {
             label1.Text = "ВЦЧ: Введены некорректные значения чисел.";
        }
    }
}

else if (command == "МЦЧ")//Множення цілих чисел: A1 = A1 × A2
{
    int a1Index = int.Parse(a1) - 1;
    int a2Index = int.Parse(a2) - 1;
```

```
                if (a1Index >= 0 && a1Index < dataGridView1.Rows.Count && a2Index >= 0 && a2Index <
dataGridView1.Rows.Count)
                {
                    string valueA1 = dataGridView1.Rows[a1Index].Cells[3].Value.ToString();
                    string valueA2 = dataGridView1.Rows[a2Index].Cells[3].Value.ToString();

                    if (int.TryParse(valueA1, out int num1) && int.TryParse(valueA2, out int num2))
                    {
                        int mno = num1 * num2;
                        if (mno == 0)
                        {
                            omegaFlag = 0;
                            zFlag = 1;
                        }
                        else if (mno < 0)
                        {
                            omegaFlag = 1;
                            zFlag = 0;
                        }
                        else
                        {
                            omegaFlag = 2;
                            zFlag = 0;
                        }
                        dataGridView1.Rows[a1Index].Cells[3].Value = mno.ToString();
                    }
                    else
                    {
                        label1.Text = "МЦЧ: Введены некорректные значения чисел.";
                    }
                }
            }

            else if (command == "ДЦЧ")//Ділення цілих чисел: A1 = A1 ÷ A2
            {
                int a1Index = int.Parse(a1) - 1;
                int a2Index = int.Parse(a2) - 1;

                if (a1Index >= 0 && a1Index < dataGridView1.Rows.Count && a2Index >= 0 && a2Index <
dataGridView1.Rows.Count)
                {
                    string valueA1 = dataGridView1.Rows[a1Index].Cells[3].Value.ToString();
                    string valueA2 = dataGridView1.Rows[a2Index].Cells[3].Value.ToString();

                    if (int.TryParse(valueA1, out int num1) && int.TryParse(valueA2, out int num2))
                    {
                        if (num2 != 0)
                        {
                            int del = num1 / num2;
                            if (del == 0)
                            {
                                omegaFlag = 0;
                                zFlag = 1;
                            }
                            else if (del < 0)
                            {
                                omegaFlag = 1;
                                zFlag = 0;
                            }
                            else
                            {
                                omegaFlag = 2;
```

```
                zFlag = 0;
            }
            dataGridView1.Rows[a1Index].Cells[3].Value = del.ToString();
        }
        else
        {
            label1.Text = "ДЦЧ: Деление на 0.";
        }
    }
    else
    {
        label1.Text = "ДЦЧ: Введены некорректные значения чисел.";
    }
  }
}

else if (command == "МОД")//Остача від ділення цілих чисел
{
  int a1Index = int.Parse(a1) - 1;
  int a2Index = int.Parse(a2) - 1;

  if (a1Index >= 0 && a1Index < dataGridView1.Rows.Count && a2Index >= 0 && a2Index <
dataGridView1.Rows.Count)
  {
    string valueA1 = dataGridView1.Rows[a1Index].Cells[3].Value.ToString();
    string valueA2 = dataGridView1.Rows[a2Index].Cells[3].Value.ToString();

    if (int.TryParse(valueA1, out int num1) && int.TryParse(valueA2, out int num2))
    {
        int mod = num1 % num2;
        if (mod == 0)
        {
            omegaFlag = 0;
            zFlag = 1;
        }
        else if (mod < 0)
        {
            omegaFlag = 1;
            zFlag = 0;
        }
        else
        {
            omegaFlag = 2;
            zFlag = 0;
        }
        dataGridView1.Rows[a1Index].Cells[3].Value = mod.ToString();
    }
    else
    {
        label1.Text = "МОД: Введены некорректные значения чисел.";
    }
  }
}

else if (command == "ВИД")//Вивід масиву дійсних чисел у кількості A2, починаючи з адреси A1
{
  int startAddress = int.Parse(a1);
  int count = int.Parse(a2);

  if (dataGridView1.Rows.Count >= startAddress + count)
  {
    string output = "";
    for (int i = startAddress; i < startAddress + count; i++)
```

```
                {
                    output += dataGridView1.Rows[i - 1].Cells[3].Value.ToString() + " ";
                }
                label1.Text = "Масив дійсних чисел: " + output;
            }
        }
        else if (command == "ВИЦ")//Вивід масиву цілих чисел у кількості A2, починаючи з адреси A1
        {
            int startRow = int.Parse(a1);
            int count = int.Parse(a2);

            if (dataGridView1.Rows.Count >= startRow + count)
            {
                string output = "";
                for (int i = startRow; i < startRow + count; i++)
                {
                    output += dataGridView1.Rows[i - 1].Cells[3].Value.ToString() + " ";
                }
                label1.Text = "Масив цілих чисел: " + output;
            }
        }

        else if (command == "БПР")//Безумовний перехід з поточного рядка на рядок A2
        {

            if (int.TryParse(a2, out int targetRow))
            {

                currentRow = targetRow - 1;
                continue;
            }
            else
            {
                label1.Text = "Некорректное значение операнда A2.";
                break;
            }
        }

        else if (command == "ПЕЦ")//Переведення цілого числа A1 у дійсне A2
        {
            int a1Index = int.Parse(a1) - 1;
            int a2Index = int.Parse(a2) - 1;

            if (a1Index >= 0 && a1Index < dataGridView1.Rows.Count && a2Index >= 0 && a2Index <
dataGridView1.Rows.Count)
            {
                string valueA1 = dataGridView1.Rows[a1Index].Cells[3].Value.ToString();
                string valueA2 = dataGridView1.Rows[a2Index].Cells[3].Value.ToString();

                int num1;
                if (int.TryParse(valueA1, out num1))
                {
                    double num2;
                    if (double.TryParse(valueA2, out num2))
                    {
                        double result = (double)num1;
                        dataGridView1.Rows[a2Index].Cells[3].Value = result.ToString();
                    }
                    else
                    {
                        label1.Text = "Некорректное значение в ячейке A2.";
                    }
                }
```

```
                else
                {
                    label1.Text = "Некорректное значение в ячейке A1.";
                }
            }
        }

        else if (command == "ПДЦ")//Переведення дійсного числа A1 в ціле A2
        {
            int a1Index = int.Parse(a1) - 1;
            int a2Index = int.Parse(a2) - 1;

            if (a1Index >= 0 && a1Index < dataGridView1.Rows.Count && a2Index >= 0 && a2Index <
dataGridView1.Rows.Count)
            {
                string valueA1 = dataGridView1.Rows[a1Index].Cells[3].Value.ToString();
                string valueA2 = dataGridView1.Rows[a2Index].Cells[3].Value.ToString();

                double num1;
                if (double.TryParse(valueA1, out num1))
                {
                    int num2;
                    if (int.TryParse(valueA2, out num2))
                    {
                        int result = (int)num1;
                        dataGridView1.Rows[a2Index].Cells[3].Value = result.ToString();
                    }
                    else
                    {
                        label1.Text = "Некорректное значение в ячейке A2.";
                    }
                }
                else
                {
                    label1.Text = "Некорректное значение в ячейке A1.";
                }
            }
        }

        else if (command == "УПЗ")//Умовний перехід: якщо прапор Z дорівнює 0 — перехід на рядок
A1,якщо прапор Z дорівнює 1 — перехід на рядок A2
        {
            if (zFlag == 1 &&  int.TryParse(a1, out int targetRow))
            {
                currentRow = targetRow - 1;
                continue;
            }
            else if (zFlag == 0 &&  int.TryParse(a2, out int TtargetRow))
            {
                currentRow = TtargetRow - 1;
                continue;
            }
        }
        else if (command == "УПЛ")//Умовний перехід: якщо прапор ω дорівнює 0 або 2 — перехід на
рядок A1, якщо прапор ω дорівнює 1 — перехід на рядок A2

        {
            if ((omegaFlag == 0 || omegaFlag == 2)  && int.TryParse(a1, out int targetRow))
            {
                currentRow = targetRow - 1;
                continue;
            }
```

```
                else if (omegaFlag == 1 && int.TryParse(a2, out int TtargetRow))
                {
                    currentRow = TtargetRow - 1;
                    continue;
                }
            }

            else if (command == "ИТР")//Зрушити A2 (задіяний зараз індекс ячійки масиву) в адресі A1 на A2
елементів
            {
                int address1 = int.Parse(a1);
                int address2 = int.Parse(a2);

                if (address1 > 0 && address1 <= dataGridView1.Rows.Count && address2 >= 0 && address2 <=
dataGridView1.Rows.Count)
                {
                    string arrayAddressValue = dataGridView1.Rows[address1 - 1].Cells[2].Value.ToString();
                    string elementsToShiftValue = dataGridView1.Rows[address2 - 1].Cells[3].Value.ToString();

                    if (int.TryParse(arrayAddressValue, out int arrayAddress) && int.TryParse(elementsToShiftValue,
out int elementsToShift))
                    {
                        int newIndex = arrayAddress + elementsToShift;
                        dataGridView1.Rows[address1 - 1].Cells[2].Value = newIndex.ToString();
                    }
                }
            }
            else if (command == "ОСТ")
            {
                break;
            }
        }
        currentRow++;
    }
}

private void button3_Click(object sender, EventArgs e)
{
    string desktopPath = Environment.GetFolderPath(Environment.SpecialFolder.Desktop);

    string baseFileName = "data.txt";

    string filePath = Path.Combine(desktopPath, baseFileName);

    if (File.Exists(filePath))
    {
        int counter = 1;
        string newFileName = Path.GetFileNameWithoutExtension(baseFileName) + "_" + counter.ToString() +
Path.GetExtension(baseFileName);
        string newFilePath = Path.Combine(desktopPath, newFileName);

        while (File.Exists(newFilePath))
        {
            counter++;
            newFileName = Path.GetFileNameWithoutExtension(baseFileName) + "_" + counter.ToString() +
Path.GetExtension(baseFileName);
            newFilePath = Path.Combine(desktopPath, newFileName);
        }

        filePath = newFilePath;
    }

    using (StreamWriter writer = new StreamWriter(filePath))
```

```
        {
            foreach (DataGridViewRow row in dataGridView1.Rows)
            {
                if (row.Cells[1].Value != null)
                {
                    string adr = row.Cells[0].Value.ToString();
                    string kom = row.Cells[1].Value.ToString();
                    string a1 = row.Cells[2].Value.ToString();
                    string a2 = row.Cells[3].Value.ToString();

                    if (kom == "ПЕР" && (a1 != "000" || a2 != "000"))
                    {
                        string line = $"{adr}: {kom} {a1} {a2}";
                        writer.WriteLine(line);
                    }
                    else if (kom != "ПЕР")
                    {
                        string line = $"{adr}: {kom} {a1} {a2}";
                        writer.WriteLine(line);
                    }
                }
            }
        }

        MessageBox.Show($"Файл успешно сохранен на рабочем столе с именем:
{Path.GetFileName(filePath)}", "Сохранение файла", MessageBoxButtons.OK, MessageBoxIcon.Information);
    }

    // Обработчик события для TextBox (ввода чисел)
    private void textBox1_TextChanged(object sender, EventArgs e)
    {
        // Ваш код для обработки события
    }

    // Обработчик события для DataGridView
    private void dataGridView1_CellContentClick(object sender, DataGridViewCellEventArgs e)
    {
        int rowCount = dataGridView1.Rows.Count;

        if (rowCount > 0 && dataGridView1.Rows[rowCount - 1].Cells[0].Value == null)
        {
            rowCount--;
        }

        for (int i = rowCount; i < 512; i++)
        {
            DataGridViewRow row = new DataGridViewRow();
            row.CreateCells(dataGridView1);
            row.DefaultCellStyle.BackColor = SystemColors.Highlight;
            row.Cells[0].Value = (i + 1).ToString();
            row.Cells[1].Value = "ПЕР";
            row.Cells[2].Value = "000";
            row.Cells[3].Value = "000";
            dataGridView1.Rows.Add(row);
        }
    }
    private void button4_Click(object sender, EventArgs e)
    {
        dataGridView1.Rows.Clear();
        label1.Text = "";
        textBox1.Text = "";

        for (int i = 0; i < 512; i++)
```

```
            {
                DataGridViewRow row = new DataGridViewRow();
                row.CreateCells(dataGridView1);
                row.DefaultCellStyle.BackColor = SystemColors.Highlight;
                row.Cells[0].Value = (i + 1).ToString();
                row.Cells[1].Value = "ПЕР";
                row.Cells[2].Value = "000";
                row.Cells[3].Value = "000";
                dataGridView1.Rows.Add(row);
            }
        }

    }
}
```