

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ВІДОКРЕМЛЕНИЙ СТРУКТУРНИЙ ПІДРОЗДІЛ
«ПОЛТАВСЬКИЙ ПОЛІТЕХНІЧНИЙ ФАХОВИЙ КОЛЕДЖ
НАЦІОНАЛЬНОГО ТЕХНІЧНОГО УНІВЕРСИТЕТУ
«ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

Циклова комісія дисциплін програмної інженерії

КУРСОВА РОБОТА

з дисципліни «Основи програмування та АМ»

на тему Італійська гра Математико

Виконав: здобувач освіти 2 курсу,
групи 24
спеціалізації «Розробка програмного
забезпечення»

Ліфтієв А.Т.

(прізвище та ініціали)

Керівник _____

(підпис)

Олійник В.В.

(прізвище та ініціали)

Полтава – 2023

ЗМІСТ

ВСТУП	3
1. ПОСТАНОВКА ЗАДАЧІ.....	4
1.1. Постановка задачі.....	4
1.2. Основні вимоги до програми	4
1.3. Алгоритм роботи програми.....	5
2. ОПИС ПРОЕКТНИХ РІШЕНЬ	7
2.1 Опис програмних засобів	7
2.2 Опис засобів розробки	7
2.3 Тестування програми	8
3. ІНСТРУКЦІЯ КОРИСТУВАЧА	10
ВИСНОВКИ.....	14
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ	15
ДОДАТОК А. ЛІСТИНГ ПРОГРАМИ	16

ВСТУП

Математико – давня гра італійського, яка потребує від гравця логічного мислення. В гру може грати необмежена кількість учасників. Вона розвиває математичні навички, творче мислення та логіку.

В ній є набір з 52 карток, на яких записані числа від 1 до 13, картки з цими числами зустрічаються по чотири рази. Гра відбувається на квадратному полі 5x5. Головна мета розставити числа так щоб отримати комбінацію яка дасть найбільшу кількість очок.

Мета виконання курсової роботи полягає у закріпленні, поглибленні та узагальненні базових теоретичних знань з програмування та їх застосуванні до комплексного вирішення конкретного завдання. Основними завданнями написання курсової роботи є:

- закріплення, поглиблення та узагальнення знань;
- набуття навичок використання основ алгоритмізації та програмування на алгоритмічних мовах високого рівня;
- проведення ґрунтовного аналізу отриманих результатів і формування змістовних висновків стосовно їх якості.

1. ПОСТАНОВКА ЗАДАЧІ

1.1. Постановка задачі

Є набір з 52 карток, на яких записані числа від 1 до 13, причому картки з кожним з цих чисел зустрічаються чотири рази. Розробити програму, яка дозволяє імітувати гру людини з комп'ютером. Є квадратне поле з 25 клітинами. Програма випадковим чином витягує будь-яку з наявних карток і видає записане на ній число. Гравець заносить це число в одну з клітин квадрата. Так триває до тих пір, поки не будуть заповнені всі клітини квадрата. Після закінчення гри заповнення відповідного квадрата оцінюється певною кількістю очок. Мета гри - розмістити числа в клітинах так, щоб набрати найбільшу кількість очок відповідно до даної таблиці:

Комбинации чисел	В ряду или столбце	По диагонали
За 2 одинаковых числа	10 очков	20 очков
За 2 пары одинаковых чисел	20 очков	30 очков
За 3 одинаковых числа	40 очков	50 очков
За 3 одинаковых числа и два других одинаковых числа	80 очков	90 очков
За 4 одинаковых числа	160 очков	170 очков
За 5 последовательных чисел, но не обязательно по порядку расположенных	50 очков	60 очков
За три раза по 1 и два раза по 13	100 очков	110 очков
За числа 1, 13, 12, 11 и 10, но не обязательно по порядку расположенных	150 очков	160 очков
За 4 единицы	200 очков	210 очков

1.2. Основні вимоги до програми

ФУНКЦІОНАЛЬНІ ВИМОГИ – це вимоги до програмного забезпечення, які описують внутрішню роботу системи, її поведінку: обчислення даних, маніпулювання даними, обробка даних та інші специфічні функції, які має виконувати система. В даному випадку, програма повинна:

- мати можливість приймати вхідні дані від користувача;
- обробляти вхідні дані і виводити результат на екран;
- мати можливість зберігати результат обробки у вихідний файл;
- мати можливість зчитувати дані з вхідного файлу і обробляти їх.

НЕФУНКЦІОНАЛЬНІ ВИМОГИ – це вимоги до програмного забезпечення, які задають критерії для оцінки якості його роботи. В даному випадку, програма повинна:

- бути написана на мові програмування C в середовищі програмування CodeBlocks;
- працювати стабільно без помилок;
- бути написана з дотриманням стандартів кодування;
- мати зрозумілий та простий інтерфейс користувача.

1.3. Алгоритм роботи програми

Запускаємо програмний продукт. Чекаємо появи меню та обираємо один з пунктів: почати, інструкція та вихід.

При виборі першого пункту, програма виведе поле 5x5, випадкове число від 1 до 13 та запропонує користувачу вибрати рядок та стовпчик куди він хоче записати це число. Так буде повторюватися до тих пір поки всі клітинки поля не будуть заповнені.

При виборі другого пункту, програма виведе інструкцію по грі.

При виборі третього пункту, програма завершить свою роботу.

Алгоритм роботи програмного продукту в консолі представлений блок-схемою (Рисунок 1.1).

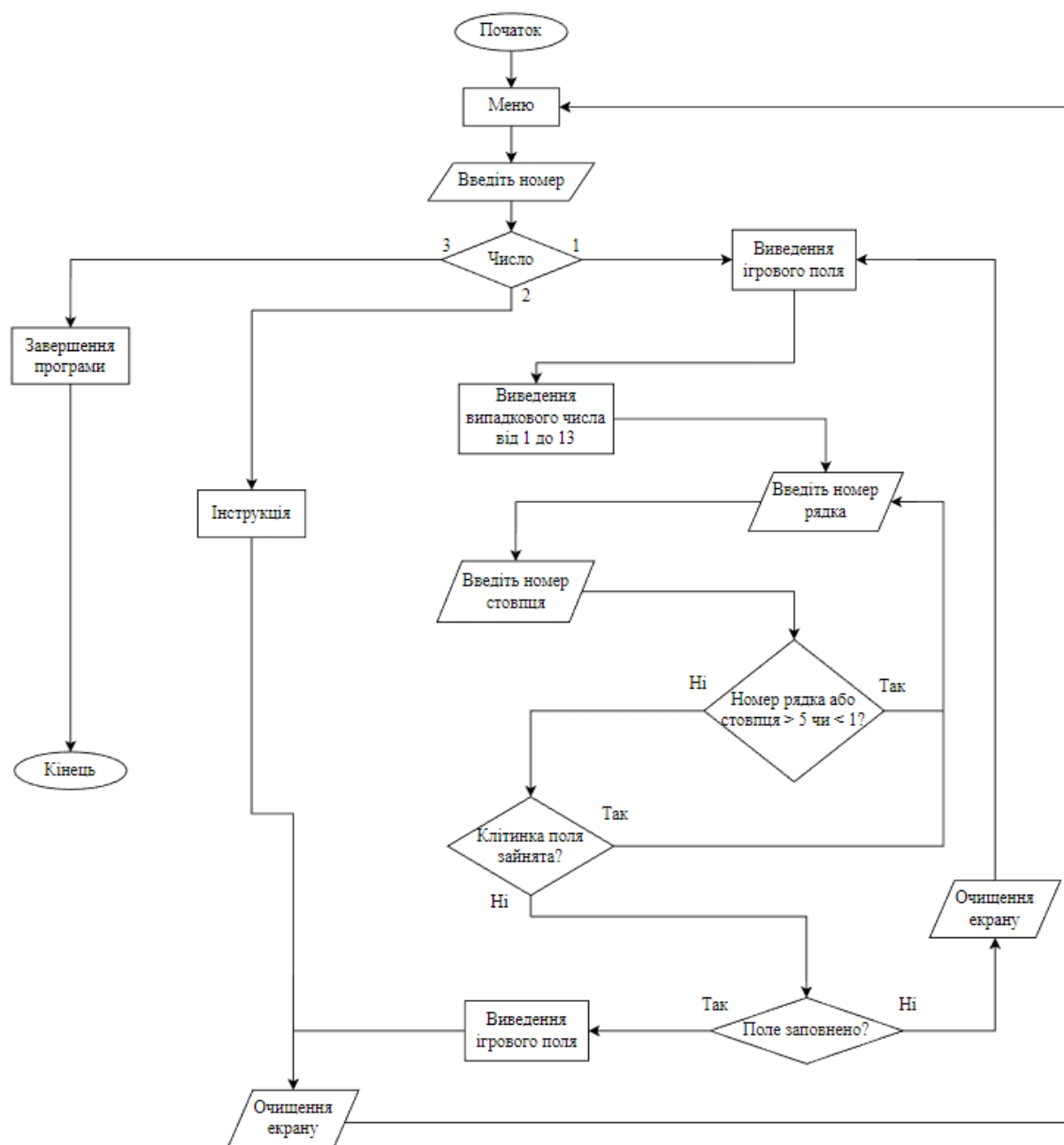


Рисунок 1.1 – Алгоритм роботи програмного продукту

2. ОПИС ПРОЕКТНИХ РІШЕНЬ

2.1 Опис програмних засобів

Мова С – універсальна, процедурна, імперативна мова програмування загального призначення, розроблена у 1972 році Деннісом Рітчі у Bell Telephone Laboratories з метою написання нею операційної системи UNIX. Хоча С і було розроблено для написання системного програмного забезпечення, наразі вона досить часто використовується для написання прикладного програмного забезпечення. С імовірно, є найпопулярнішою у світі мовою програмування за кількістю вже написаного нею програмного забезпечення, доступного під вільними ліцензіями коду та кількості програмістів, котрі її знають. Версії компіляторів для мови С існують для багатьох операційних систем та апаратних архітектур. С здійснила великий вплив на інші мови програмування, особливо на С++, яку спочатку проєктували як розширення для С, а також на Java та С#, які запозичили у С синтаксис.

ПЕРЕВАГИ:

- ефективність;
- переносимість;
- широке застосування.

НЕДОЛІКИ:

- обмежена підтримка ООП;
- обмежена стандартна бібліотека;
- вразливість до помилок.

2.2 Опис засобів розробки

Code::Blocks – вільне багатоплатформне середовище розробки програмного забезпечення. Code::Blocks написане на С++ і використовує бібліотеку wxWidgets. Маючи відкриту архітектуру, може масштабуватись за рахунок додаткових модулів. Підтримує мови програмування С, С++, D та Fortran. Code::Blocks є популярним інструментом для програмування,

особливо для початківців та середньо-досвідчених розробників. Він надає зручність та функціональність, необхідні для розробки програм у різних мовах програмування.

ПЕРЕВАГИ:

- відкритий вихідний код;
- кросплатформовість;
- легкість використання;
- розширюваність.

НЕДОЛІКИ:

- обмежені можливості редагування;
- швидкість;
- оновлення та підтримка;
- документація.

2.3 Тестування програми

Для тестування програмного продукту було обрано 4 методи тестування: метод чорної, білої, сірої скриньки та відмови.

Тестування чорної скриньки – це метод тестування програмного забезпечення, при якому перевіряється робота програми без знання її внутрішньої побудови та схеми роботи. Іншими словами, не маючи доступу до коду програми.

Тестування білої скриньки – це метод тестування програмного забезпечення, який заснований на аналізі внутрішньої структури програми та її компонентів. У цьому методі тестування тестувальник має повний доступ до вихідного коду, архітектури, алгоритмів та інших деталей програмного продукту.

Тестування сірої скриньки – це метод тестування, який поєднує підходи білої та чорної скриньки. Тестувальник має обмежений доступ до внутрішньої структури програми, такий як документація або обмежений

вихідний код. Це дозволяє здійснювати більш цілеспрямоване тестування, використовуючи знання про внутрішні механізми програми.

Тестування відмови – це метод тестування, що передбачає активне створення відмов або помилок в програмі з метою виявлення реакції системи на неправильні дії або ситуації. Тестувальник намагається викликати помилки, надмірне навантаження або некоректну поведінку, щоб оцінити, як програма справляється з такими ситуаціями.

Під час тестування було виявлено, що програма працює некоректно після введення чогось окрім цифр (Рисунок 2.1).

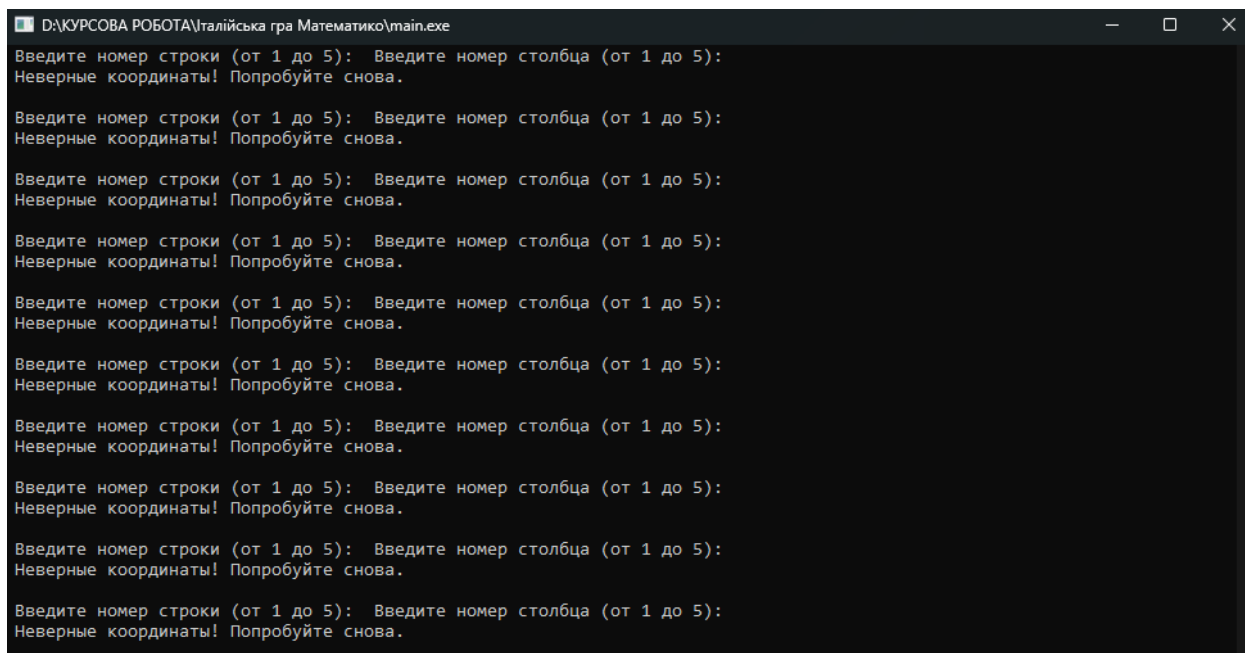


Рисунок 2.1 – Програма після введення некоректних даних

3. ІНСТРУКЦІЯ КОРИСТУВАЧА

Програма створена для гри в Математико.

Вимоги до системи: процесора, розміру оперативної пам'яті, версії операційної систем, встановлені драйвера – мінімальні.

Запуск виконується за допомогою файлу main.exe.

Після запуску програмного продукту перед користувачем з'явиться головне меню (Рисунок 3.1)

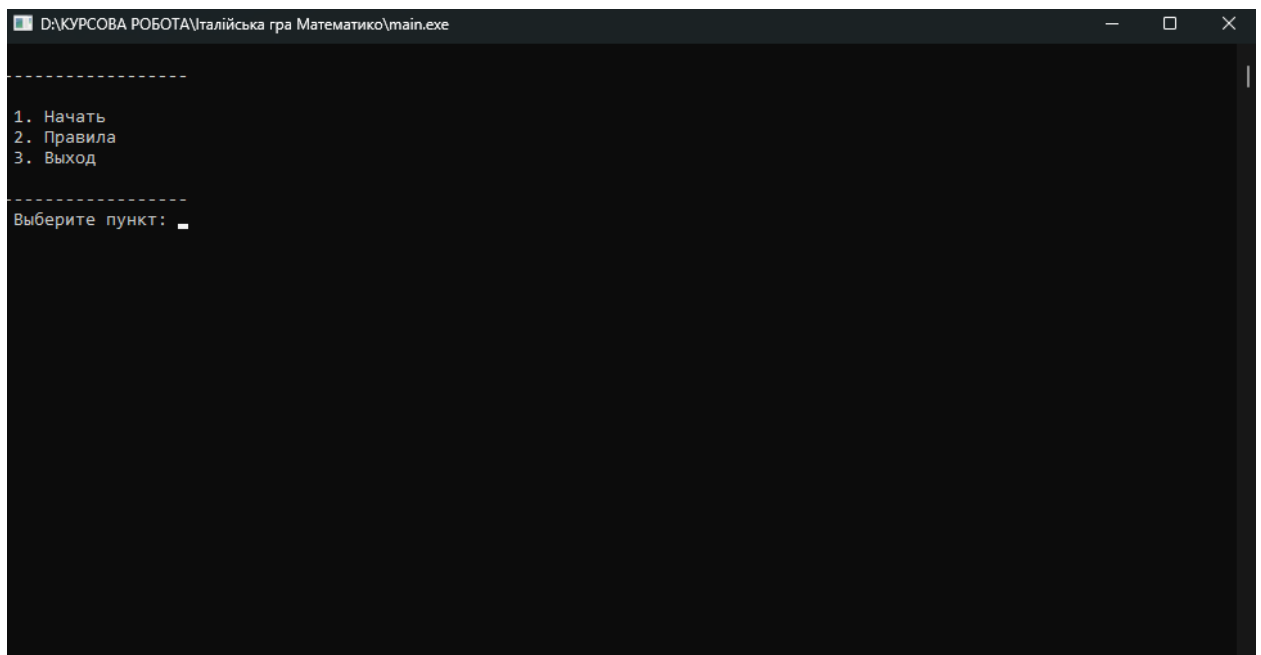


Рисунок 3.1 – Меню програми

Програма попросить обрати пункт 1, 2 чи 3.

При виборі пункту 1 почнеться гра.

Програма виведе поки що пусту таблицю, випадкове число від 1 до 13 та попросить користувача ввести номер спочатку рядка, а потім стовпця куди він хоче записати це випадкове число (Рисунок 3.2).

В разі неправильного введення даних програма виведе повідомлення в якому йдеться де саме була допущена помилка та попросить ввести дані ще раз (Рисунок 3.3).

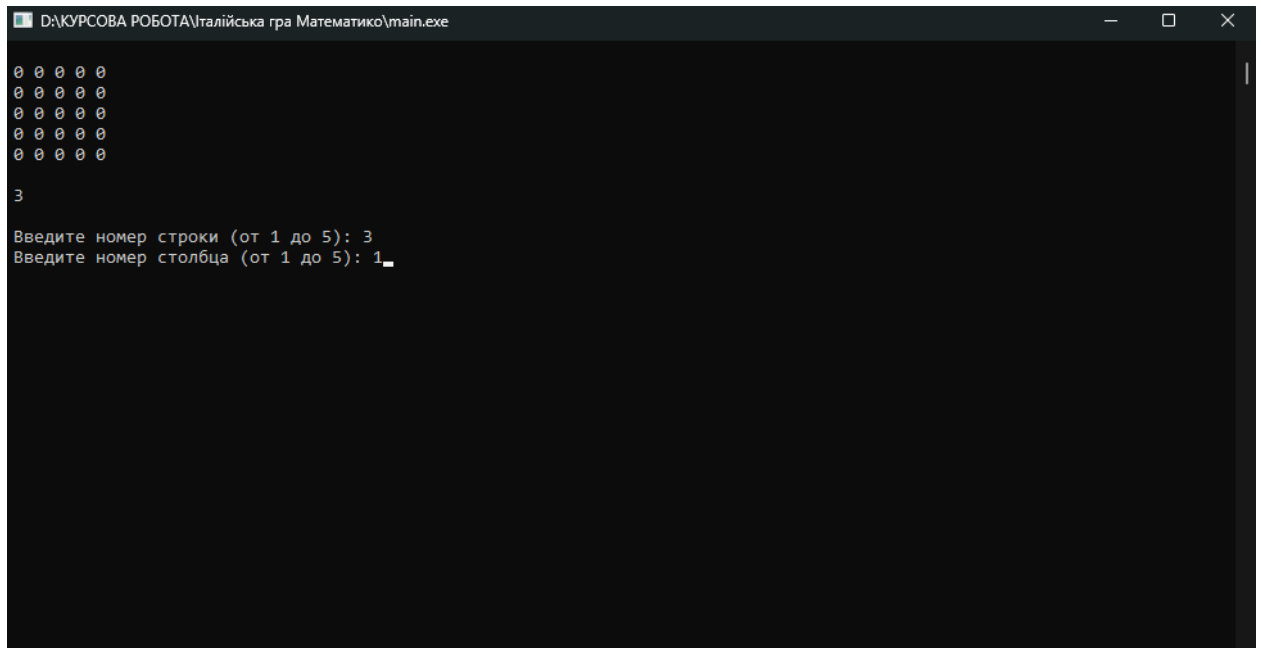


Рисунок 3.2 – Введення даних з клавіатури в консоль

```
Введіть номер строки (от 1 до 5): -4
Введіть номер столбца (от 1 до 5): 23
Неверные координаты! Попробуйте снова.
```

Рисунок 3.3 – Програма видала помилку через неправильні дані

Після того як користувач введе номер рядка та стовпця програма знову виведе таблицю вже з введеним числом, нове випадкове число від 1 до 13 та ще раз попросить ввести координати поля. Так буде повторюватись до тих пір поки все поле не буде заповнено (Рисунок 3.3).

Потім програма останній раз виведе заповнене поле, таблицю для підрахунку очок та напише, що для того щоб повернутися у меню потрібно натиснути Enter (Рисунок 3.4).

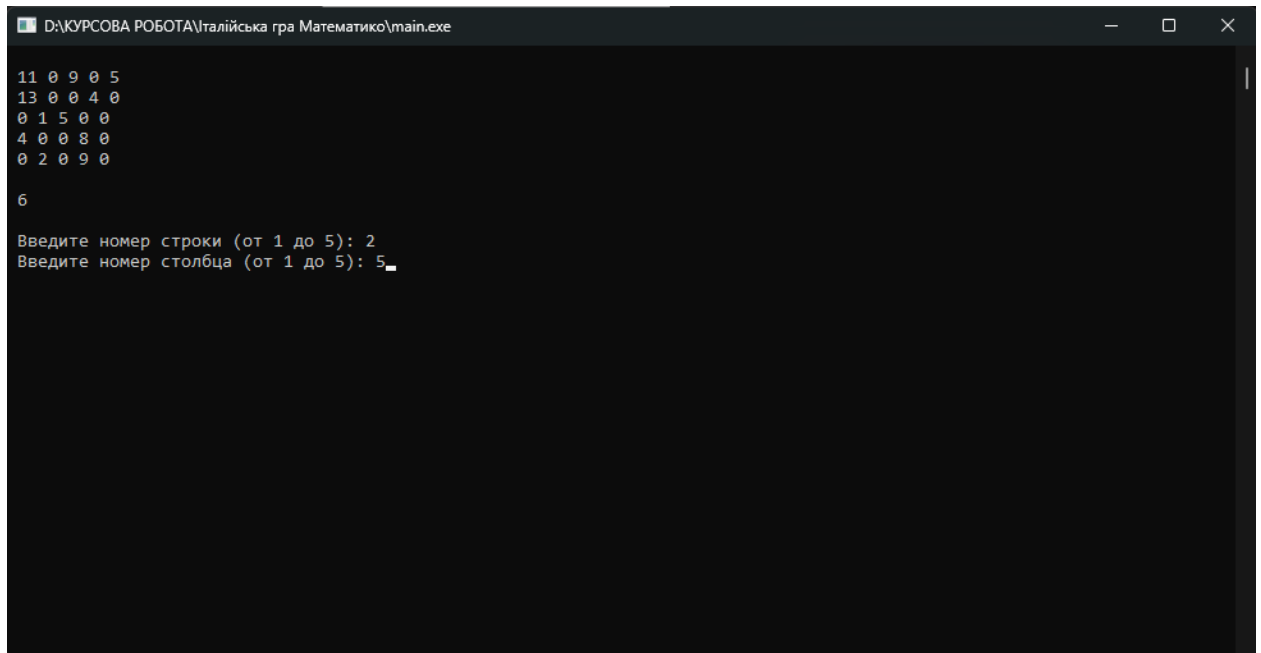


Рисунок 3.3 – Процес заповнення ігрового поля

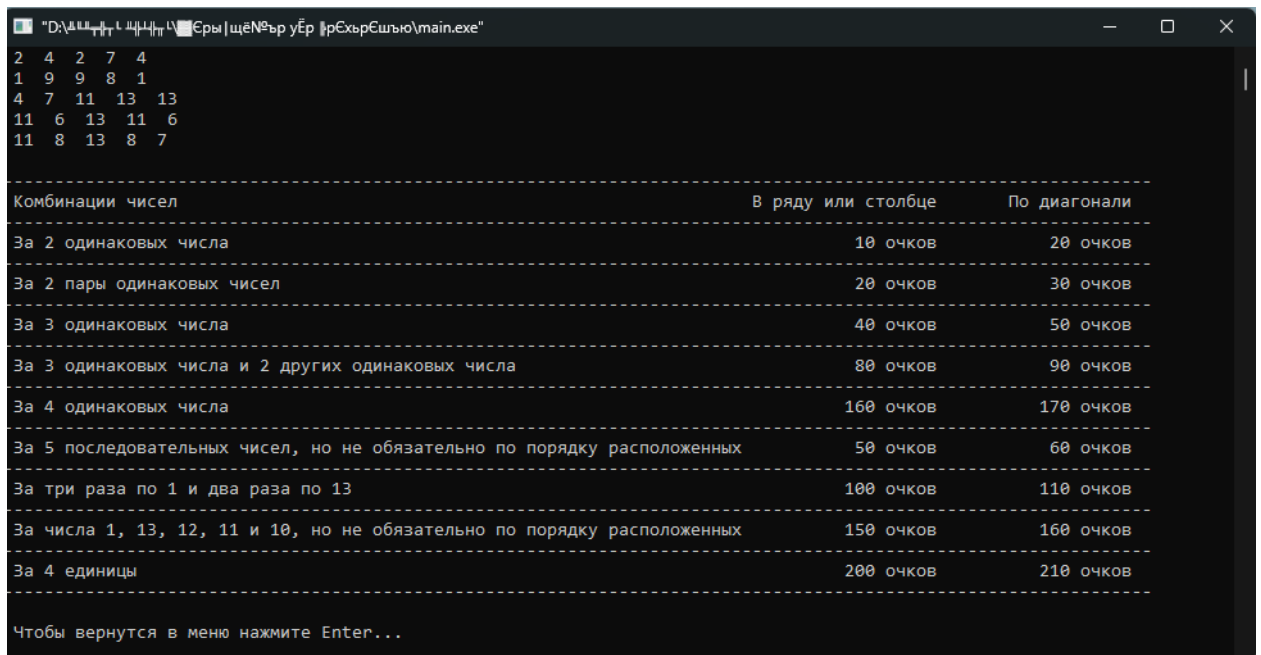


Рисунок 3.4 – Заповнене ігрове поле

При виборі пункту 2 програма виведе інструкцію з гри та скаже натиснути Enter щоб повернутися у меню (Рисунок 3.5).

При виборі пункту 3 програма завершить свою роботу.

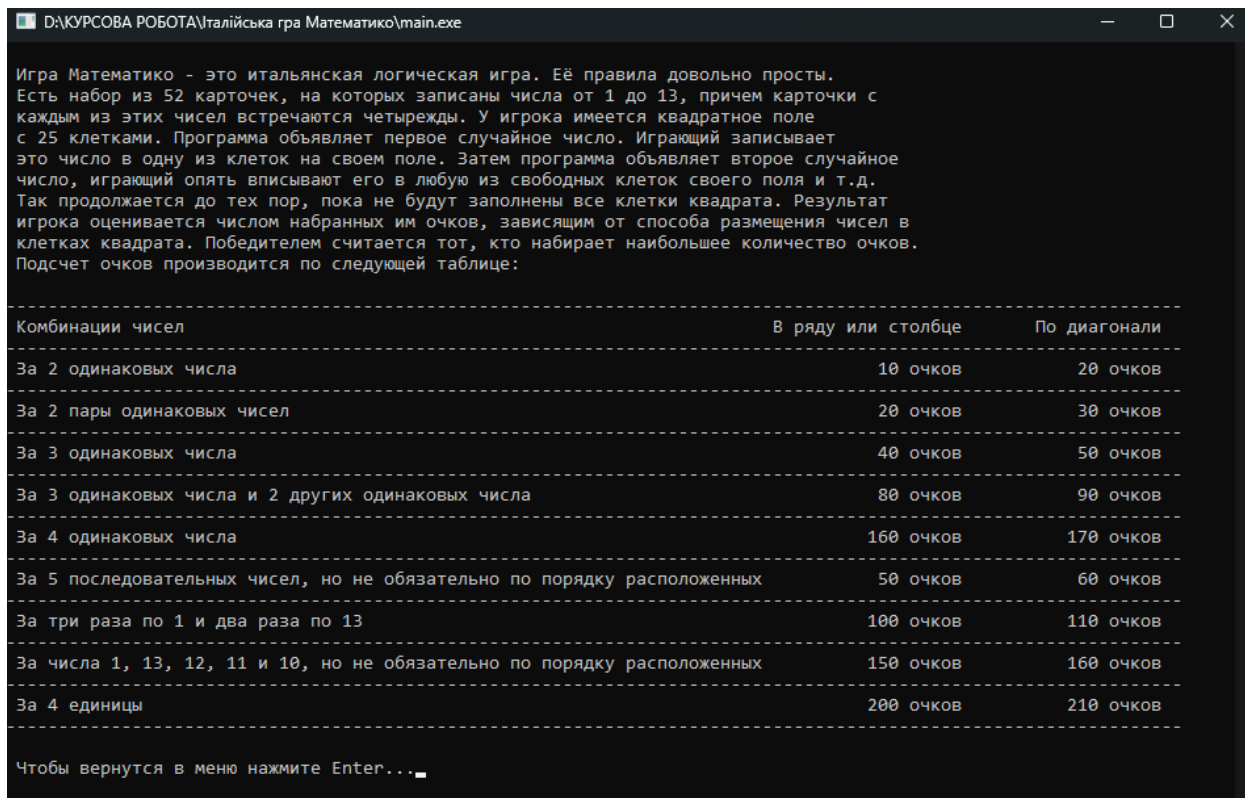


Рисунок 3.5 – Інструкція з гри

ВИСНОВКИ

В ході написання курсової роботи було створено програмний продукт. Його суть імітувати італійську гру Математико. Програма написана на мові C в середовищі програмування CodeBlocks. В програмному продукті реалізовані можливості:

- вводу та виводу на екран;
- аналіз вхідних даних;
- створення грального поля та запис в нього даних;
- взаємодія з користувачем через консоль.

В процесі виконання курсової роботи:

- на етапі написання коду було вдосконалено навички з програмування на мові C;
- на аналітичному етапі проектування програмного забезпечення вдосконалено навички з побудови блок-схем;
- на етапі експериментального дослідження отримано результати роботи програмного коду в різних режимах;
- на етапі аналізу отриманих результатів оброблено всі вихідні дані.

Під час написання курсової роботи було закріплено теоретичні знання та вдосконалено практичні навички.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. МЕТОДИЧНІ ВКАЗІВКИ ДО ВИКОНАННЯ КУРСОВИХ РОБІТ ДЛЯ ЗДОБУВАЧІВ ОСВІТИ СПЕЦІАЛЬНОСТІ 121 «ІНЖЕНЕРІЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ»: МЕТОДИЧНІ ВКАЗІВКИ / Відокремлений структурний підрозділ «Полтавський політехнічний фаховий коледж Національного технічного університету «Харківський політехнічний інститут»; [уклад.: В.В.Олійник]. – Полтава: ВСП ППФК НТУ «ХПІ», 2022. – 22 с.
2. Браян У. Керніган, Деніс М. Річі. Мова програмування C, 2-е видання, 2011. – 232с.
3. З. Шпак. Програмування мовою C. – Львів: Оріяна-Нова, 2006. – 432с.: іл.
4. Мартін Роберт. Чистий код: створення і рефакторинг за допомогою Agile / Пер. з англ. І. Бондар-Терещенко. – Харків : Вид-во «Ранок» : Фабула, 2019. – 448 с.
5. Програмування на C/C++ [Електронний ресурс] – Режим доступу до ресурсу: <https://www.eolymp.com/uk/blogs/posts/26>
6. Мова програмування C [Електронний ресурс] – Режим доступу до ресурсу: <http://centertc.org/uk/courses/103-c-programming-language.html>
7. Code::Blocks [Електронний ресурс] – Режим доступу до ресурсу: <https://www.codeblocks.org>
8. Chat GPT [Електронний ресурс] – Режим доступу до ресурсу: <https://chat.openai.com>

ДОДАТОК А. ЛІСТИНГ ПРОГРАМИ

```

#include <stdio.h>
#include <locale.h>

int main()
{
    char *locale = setlocale(LC_ALL, "");

    while (1){
        printf("\n-----\n");
        printf("\n 1. Начать\n");
        printf("\n 2. Правила\n");
        printf("\n 3. Выход\n");

        int num;
        printf("\n-----");
        printf("\n Выберите пункт: ");
        scanf("%d", &num);

        system("cls");

        int numbers[13] = { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13};
        int count[13] = {0};

        int game_field[5][5] = {0};
        int num_field = 0;

        int row, col;

        switch(num){
            case 1:
                srand(time(NULL));

                while (num_field < 25){

                    printf("\n");
                    for (int i = 0; i < 5; i++) {
                        printf("\n");
                        for (int j = 0; j < 5; j++) {
                            printf("   %d", game_field[i][j]);
                        }
                        printf("\n");
                        printf("\n");
                    }

                    int ind_num = rand() % 13;

```



```

if (count[ind_num] < 4) {
    printf("\n %d\n", numbers[ind_num]);
    count[ind_num]++;
}

int num1 = ind_num;

while (1){
    printf("\n Введите номер строки (от 1 до 5): ");
    scanf("%d", &row);
    printf(" Введите номер столбца (от 1 до 5): ");
    scanf("%d", &col);

    if (row < 1 || row > 5 || col < 1 || col > 5) {
        printf("\n Неверные координаты! Попробуйте снова.\n");
        continue;
    } else if (game_field[row-1][col-1] > 0) {
        printf("\n Выбранная ячейка уже занята!\n");
        continue;
    } else {
        break;
    }
}

game_field[row-1][col-1] = num1 + 1;
num_field++;

system("cls");
}

printf("\n");
for (int i = 0; i < 5; i++) {
    printf("\n");
    for (int j = 0; j < 5; j++) {
        printf("    %d ", game_field[i][j]);
    }
    printf("\n");
    printf("\n");
}

printf("\n-----");
printf("\n Комбинации чисел                В ряду или столбце    По диагонали");
printf("\n-----");
printf("\n За 2 одинаковых числа                10 очков        20 очков");
printf("\n-----");
printf("\n За 2 пары одинаковых чисел            20 очков        30 очков");

```

```

printf("\n-----");
printf("\n 3а 3 одинаковых числа                40 очков    50 очков");
printf("\n-----");
printf("\n 3а 3 одинаковых числа и 2 других одинаковых числа                80 очков    90 очков");
printf("\n-----");
printf("\n 3а 4 одинаковых числа                160 очков    170 очков");
printf("\n-----");
printf("\n 3а 5 последовательных чисел, но не обязательно по порядку расположенных    50 очков    60
очков");
printf("\n-----");
printf("\n 3а три раза по 1 и два раза по 13                100 очков    110 очков");
printf("\n-----");
printf("\n 3а числа 1, 13, 12, 11 и 10, но не обязательно по порядку расположенных    150 очков    160
очков");
printf("\n-----");
printf("\n 3а 4 единицы                200 очков    210 очков");
printf("\n-----\n");

printf("\n Чтобы вернуться в меню нажмите Enter...");

getch();
system("cls");
break;

```

case 2:

```

printf("\n  Игра Математико - это итальянская логическая игра. Её правила довольно просты.");
printf("\n  Есть набор из 52 карточек, на которых записаны числа от 1 до 13, причем карточки с");
printf("\n  каждым из этих чисел встречаются четырежды. У игрока имеется квадратное поле");
printf("\n  с 25 клетками. Программа объявляет первое случайное число. Играющий записывает");
printf("\n  это число в одну из клеток на своем поле. Затем программа объявляет второе случайное");
printf("\n  число, играющий опять вписывают его в любую из свободных клеток своего поля и т.д.");
printf("\n  Так продолжается до тех пор, пока не будут заполнены все клетки квадрата. Результат");
printf("\n  игрока оценивается числом набранных им очков, зависящим от способа размещения чисел в");
printf("\n  клетках квадрата. Победителем считается тот, кто набирает наибольшее количество очков.");
printf("\n  Подсчет очков производится по следующей таблице:\n");
printf("\n-----");
printf("\n Комбинации чисел                В ряду или столбце    По диагонали");
printf("\n-----");
printf("\n 3а 2 одинаковых числа                10 очков    20 очков");
printf("\n-----");
printf("\n 3а 2 пары одинаковых чисел                20 очков    30 очков");
printf("\n-----");
printf("\n 3а 3 одинаковых числа                40 очков    50 очков");
printf("\n-----");
printf("\n 3а 3 одинаковых числа и 2 других одинаковых числа                80 очков    90 очков");
printf("\n-----");
printf("\n 3а 4 одинаковых числа                160 очков    170 очков");

```

```

printf("\n-----");
printf("\n За 5 последовательных чисел, но не обязательно по порядку расположенных      50 очков      60
очков");
printf("\n-----");
printf("\n За три раза по 1 и два раза по 13      100 очков      110 очков");
printf("\n-----");
printf("\n За числа 1, 13, 12, 11 и 10, но не обязательно по порядку расположенных      150 очков      160
очков");
printf("\n-----");
printf("\n За 4 единицы      200 очков      210 очков");
printf("\n-----\n");

printf("\n Чтобы вернуться в меню нажмите Enter...");

getch();
system("cls");
break;

case 3:
    exit(0);

default:
    printf("\n Неверная операция\n");
    getch();
    system("cls");
    break;
}
}

getch();
return 0;
}

```