

Федеральное агентство железнодорожного транспорта
Омский государственный университет путей сообщения

Кафедра «Автоматика и системы управления»

ПРОГРАММИРОВАНИЕ ИГРЫ НА ЯЗЫКЕ С
Пояснительная записка к курсовой работе
по дисциплине «Программирование»
ИНМВ.400012.000 ПЗ

Студент гр. 21 м

_____ К.Н. Юрукина
«1» июня _____ 2022 г.

Руководитель –
доцент кафедры АиСУ
_____ А.В. Пономарев
«1» июня _____ 2022 г.

Омск 2022

Реферат

УДК 004.42

Пояснительная записка к курсовой работе содержит 21 страниц, 16 рисунков, 3 использованных источника, 2 приложения.

Объектом курсовой работы является консольная игра «Быки и коровы».

Цель курсовой работы – получение основных навыков использования языка Си, создание игры с искусственным интеллектом.

Результатом курсовой работы является игра «Быки и коровы», написанная на языке Си в программе Visual Studio Code.

В процессе создания игры была изучена лексика языка Си.

Пояснительная записка выполнена в текстовом редакторе Microsoft Word 2010.

Введение

Язык программирования Си отличается минимализмом. Си создавался с одной важной целью: сделать более простым написание больших программ с минимумом ошибок по правилам процедурного программирования, не добавляя на итоговый код программ лишних накладных расходов для компилятора.

Си предлагает следующие важные особенности: простую языковую базу, из которой вынесены в библиотеки многие существенные возможности; ориентацию на процедурное программирование, обеспечивающую удобство применения структурного стиля программирования; систему типов; использование препроцессора; непосредственный доступ к памяти компьютера через использование указателей; минимальное число ключевых слов; передачу параметров в функцию по значению, а не по ссылке; указатели на функции и статические переменные, структуры и объединения; средства объектно-ориентированного программирования.

Часть отсутствующих возможностей относительно легко имитируется встроенными средствами, часть добавляется с помощью сторонних библиотек, часть реализуется в некоторых компиляторах в виде расширений языка.

Язык Си остаётся языком, реализованным на максимальном количестве аппаратных платформ, и одним из самых популярных языков программирования, особенно в мире свободного программного обеспечения.

Курсовая работа является примером использования языка Си.

Содержание

Введение	3
1 Информация об игре	5
2 Реализация игры	6
3 Инструкция пользователя	9
Заключение	12
Библиографический список	13
Приложение А	14
Приложение Б.....	21

1 Информация об игре

1.1 Правила игры

Быки и коровы – это логическая игра, рассчитанная на двух игроков, целью каждого из которых является отгадать число противника.

Каждый игрок загадывает четырехзначное число без повторяющихся цифр. Игроки должны найти четыре быка. Если в веденном числе совпала цифра и ее место, то это бык. Если только цифра, то корова. У игроков неограниченное количество попыток, но выигрывает тот, кто соберет четырех быков раньше противника.

1.2 История игры

Она придумана сравнительно недавно, в 70-е годы, завоевала огромную популярность во многих странах. Ее наиболее распространенный вариант выпускается в виде комплекта под названием «Mastermind» (мастермайнд, буквальный перевод – «выдающийся ум»). В комплекте «Mastermind» роль цифр выполняют колышки шести цветов (красные, желтые, синие, зеленые, белые, черные), они вставляются в отверстия доски.

1.3 Советы

Идеи теории информации полезные для улучшения показателей в игре:

- вопрос нужно задавать так, чтобы ответ на него давал максимальное количество информации;
- для этого вопрос нужно формировать так, чтобы вероятности различных ответов были по возможности близкими;
- кроме того, вопрос должен быть таким, чтобы ответ на него не содержал информацию, полученную ранее из предыдущих вопросов.

2 Реализация игры

В программе используются библиотеки:

- #include <stdio.h> // библиотека ввода–вывода
- #include <string.h> // библиотека для работы со строками
- #include <stdlib.h> // библиотека со вспомогательными функциями
- #include <time.h> // библиотека даты и времени

Программа разработана с использованием структурного подхода к программированию. Для решения поставленной задачи были реализованы несколько функций.

Все начинается с функции void main(). Она не принимает и ничего не передает, но после вывода правил игры, обращается к другим функциям. Вначале нас спрашивают о режиме игры (0 – продолжить (появляется только при наличии файла «game.txt»), 1 – одиночный, 2 – с другом). Для начала рассмотрим одиночный режим.

Тогда функция main() обращается к void game_one(). В цикле с помощью рандома формируется четырехзначное, в нем же оно и проверяется в функции int pr(), проверяющая число на наличие повтора цифр. Далее программа попадает в другой цикл, в котором уже происходят основные действия игры. В условии он проверяет целочисленную переменную «w», являющейся полем структуры и использующейся, как булева переменная, заданной сразу после библиотек. Если она равна 1, это значит, что игрок отгадал число. В цикле программа обращается к функциям:

- void perenos (struct data *pl),
- void input (int i, struct data *pl),
- int check (char a[]),
- int pr_num (char *a),
- void score_bk (int a, int n, struct data *pl, char *guess),
- void print_table_one (struct data *pl).

К функции void perenos (struct data *pl), принимающей ссылку на структуру, программа обращается только тогда, когда игрок совершает больше 10 попыток. Данные массива char number[10][5], который состоит в структуре, переносятся на один элемент вперед, а в пустое, образовавшееся место, в которое записывается новое число в функции void input (int i, struct data *pl).

Функция void input (int i, struct data *pl) принимает переменную «i», которая играет роль булевой, и ссылку на структуру. Эта функция отвечает за запись введенного игроком числа. «i» нужна лишь для того, чтобы, если это одиночная игра, не выводить обращение игроку или, если совместный режим, выводить.

Функция int check (char a[]) отвечает за правильное написание числа, то есть, чтобы оно состояло только из четырех цифр. Принимает проверяемый символьный массив и передает 1, если проверка не прошла, и 0, если наоборот.

Функция `int pr_num (char *a)` отвечает за правильное содержание числа, то есть проверяет число на повтор цифр в нем. Аргументом является ссылка на это число. Если цифры повторялись, то функция передает 0, иначе 1.

Функция `void score_bk (int a, int n, struct data *pl, char *guess)` подсчитывает количество быков и коров. Первый аргумент – переменная, отвечающая за необходимость переноса подсчета на последнее число, в данном случае она равна 1, второй аргумент – количество попыток, третий – ссылка на структуру игрока, а четвертый – на загаданное компьютером число (в совместном режиме четвертый аргумент – ссылка число, загаданное другим игроком). В цикле программа считает отдельно все совпадения (быков и коров) и только быков. При подсчете коров из всех совпадений вычитается быки. В этой же функции делается проверка на выигрыш. Если быков четыре, в переменную «w», описанную выше, записывается 1.

Функция `void print_table_one (struct data *pl)` выводит таблицу попыток. Единственный аргумент – это ссылка на структуру с данными игрока. На экране выводятся максимум 10 попыток, напротив каждого числа изображено количество быков и коров. Благодаря функции `void perenos (struct data *pl)` игрок может использовать неограниченное количество попыток отгадать число компьютера.

Цикл функции `void game_one()` не завершится, пока игрок не соберет четыре быка. После чего у него будет выбор: начать заново в этом режиме или выбрать другой. При выборе второго ответа, программа переходит в функцию `void main()`.

В совместном режиме, в функции `void game_friend (int ex)`, все схоже за исключением нескольких моментов. Аргументом `void game_friend (int ex)` является переменная, отвечающая за продолжение незаконченной игры. Если игроки решат продолжить, то программа обратится к `void old_game (struct data *pl_1, struct data *pl_2)`.

Функция `void old_game (struct data *pl_1, struct data *pl_2)` принимает в качестве аргументов структуру каждого из игроков. Она записывает в них данные записанные в файле: количество попыток, загаданные числа и сами числа, но не количество быков и коров. Они просчитываются уже после чтения в функции `void score_bk (int a, int n, struct data *pl, char *guess)`, где переменная «a» уже равна 0.

Далее программа входит в цикл игры, который схож с циклом из предыдущего режима, но только для двух игроков. После каждого хода игроков происходит сохранение данных в функции `void save (struct data *pl_1, struct data *pl_2)`, аргументами которой являются ссылки на две структуры. В ней перезаписывается файл с уже существующими данными или создается с нуля. Файл в этой функции открывается и в ней же закрывается, чтобы в нужном случае ее удалить.

Также она отличается функцией `void print_table(struct data *pl_1, struct data *pl_2)`, принимающей в качестве аргументов ссылки на структуры игроков.

На экране выводятся попытки уже двух игроков. Их также можно вводить неограниченное количество раз.

После выигрыша одного из игроков или ничьи, то есть после выхода из цикла, файл удаляется. Потом у игроков спрашивается желание продолжить в выбранном режиме.

При положительном ответе игра начнется заново, создастся файл, который будет существовать, пока кто-нибудь не выиграет или его не удалят вручную вне игры.

Иначе уже попав в функцию `main()`, программа спросит о желании играть в принципе. При положительном ответе, спросит о режиме игры, при отрицательном – просто завершит работу программы.

После каждого `scanf` выполняется проверка на правильность ввода. В случае некорректно введенной информации, высветится сообщение об ошибке и тогда можно ввести данные еще раз.

3 Инструкция пользователя

В начале игры на экране высвечивается приветствие, правила и список режимов. Если до запуска программы в режиме «с другом» игра была не закончена, то выведется список как на рисунке 1, иначе как на рисунке 2.

```
G:\Курсовая\Код>a
Добрый день! Это игра "Быки и коровы!"
Правила игры:
игроки загадывают четырехзначное число без повтора цифр.
Потом они задают друг другу другие такие четырехзначные числа, какие на их взгляд у противника.
Если совпало место и число, то это бык (Б). Если совпало только число, то это корова (К).

Выберите режим:
0 - Продолжить
1 - одиночный
2 - с другом
█
```

Рисунок 1 – Начало игры с возможностью продолжить старую

```
G:\Курсовая\Код>a
Добрый день! Это игра "Быки и коровы!"
Правила игры:
игроки загадывают четырехзначное число без повтора цифр.
Потом они задают друг другу другие такие четырехзначные числа, какие на их взгляд у противника.
Если совпало место и число, то это бык (Б). Если совпало только число, то это корова (К).

Выберите режим:
1 - одиночный
2 - с другом
█
```

Рисунок 2 – Начало игры

Когда пользователь выбрал одиночный режим, то на экране появится сообщение о том, что число загадано и можно играть (рисунок 3).

```
Компьютер загадал число. Попробуйте отгадать!
Введите число: █
```

Рисунок 3 – Начало игры в одиночном режиме

На рисунке 4 изображен процесс игры в одиночном режиме.

	Попытки
1	1256 0Б 2К
2	9632 0Б 2К
3	0258 0Б 3К
4	2085 0Б 3К
5	8507 0Б 2К
6	8420 2Б 0К

Введите число: █

Рисунок 4 – Процесс игры в одиночном режиме

Если человек введет число некорректно, то выскочит оповещение об ошибке (рисунок 5). Так будет продолжаться, пока число не будет введено правильно.

На экране видны максимум десять последних попыток.

```
Введите число: 0000
Данные введены некорректно!
Введите число: 2336
Данные введены некорректно!
Введите число: 12369
Данные введены некорректно!
Введите число: 2
Данные введены некорректно!
Введите число: █
```

Рисунок 5 – Проверка некорректного ввода

Когда человек отгадает загаданное компьютером число, на экране будет сообщение о выигрыше и вопрос о продолжении (рисунок 6). В случае, если пользователь решит продолжить, то игра начнется с момента, как на рисунке 7.

```
      | Попытки
==|=====
1| 5623 0Б 2К
2| 9632 0Б 3К
3| 6349 2Б 1К
4| 6394 3Б 0К
5| 6294 2Б 0К
6| 6391 3Б 0К
7| 6392 3Б 0К
8| 6397 4Б 0К

Поздравляем! Вы выиграли!
Количество ходов: 8

Продолжить в этом режиме?
1 - да, 0 - нет
█
```

Рисунок 6 – Выигрыш в одиночном режиме

```
Продолжить в этом режиме?
1 - да, 0 - нет
0
Желаете сыграть еще?
1 - да, 0 - нет
█
```

Рисунок 7 – Вопрос при выходе из режима

Для совместного режима ситуация похожая.

Сначала игроки по очереди загадывают друг другу числа (рисунок 8), а том пытаются их отгадать (рисунок 9).

Игрок 1, введите загаданное число: 1234█

Рисунок 8 – Введение загаданного числа

```
      | Игрок 1 | Игрок 2 |
==|=====|=====|==
1| 1259 0Б 1К | 3598 0Б 1К | 1
2| 6325 0Б 2К | 1874 2Б 0К | 2
3| 0359 0Б 3К | 3268 1Б 1К | 3
4| 1094 0Б 2К | 1803 1Б 1К | 4

Игрок 1
Введите число: █
```

Рисунок 9 – Процесс игры в совместном режиме

Когда хотя бы один из игроков угадывает число, игра заканчивается (рисунок 10) с возможностью начать новую.

	Игрок 1	Игрок 2	
==	=====	=====	==
1	1259 0Б 1К	3598 0Б 1К	1
2	6325 0Б 2К	1874 2Б 0К	2
3	0359 0Б 3К	3268 1Б 1К	3
4	1094 0Б 2К	1803 1Б 1К	4
5	9630 4Б 0К	1234 4Б 0К	5

Ничья!
Количество ходов: 5

Числа Игроков: 1234
Продолжить в этом режиме?
1 - да, 0 - нет

Рисунок 10 – Окончание игры в совместном режиме

Если есть возможность продолжить старую игру, то игра начинается с того момента, на котором закончили.

Заключение

Во время выполнения курсовой работы были изучены принципы программирования на языке Си, были освоены основные концепции и стратегии по созданию консольного приложения «Быки и коровы», которое является результатом выполнения курсовой работы.

Были повторены темы по работе с файлами, с помощью которых была реализована функция сохранения результатов каждого из игроков. Также была изучена новая тема «Псевдослучайные числа» и библиотека «time.h», позволившие создать подбор числа, который «загадывает» компьютер.

Библиографический список

- 1 Керниган Б. Язык программирования Си 2-е издание / Б. Керниган Д. Ричи М.:Издательский дом "Вильямс", 2012. 272 с.
- 2 Википедия. Свободная энциклопедия [Электронный ресурс] / Режим доступа: [https://ru.wikipedia.org/wiki/Быки_и_коровы_\(логическая_игра\)](https://ru.wikipedia.org/wiki/Быки_и_коровы_(логическая_игра))
- 3 Википедия. Свободная энциклопедия [Электронный ресурс] / Режим доступа: [https://ru.wikipedia.org/wiki/Си_\(язык_программирования\)](https://ru.wikipedia.org/wiki/Си_(язык_программирования))

Приложение А
(обязательное)
Код программы

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <time.h>
// Данные для одного игрока
struct data {
    int n;                // Количество ходов
    char guess[5];        // Загаданное число
    char number[10][5];   // Запись попыток
    int k[10], b[10];     // Число коров и быков для каждой попытки
    int w;                // Наличие выигрыша
};
//количество быков и коров
void score_bk(int a, int n, struct data *pl, char *guess){
    int i, j;
    if (a){
        if (pl->n < 9) n = pl->n;
        else n = 9;
    }
    pl->b[n] = 0;
    pl->k[n] = 0;
    for (i = 0; i < 4; i++){
        for (j = 0; j < 4; j++){
            if (pl->number[n][i] == guess[j]){
                pl->k[n]++;
                if (i == j){
                    pl->b[n]++;
                }
            }
        }
    }
    pl->k[n] -= pl->b[n];
    if (a) pl->n++;
    if (pl->b[n] == 4){
        pl->w = 1;
    }
}
// проверка ввода числа
int pr_num(char *a){
    int i, j;
    for(i = 0; i < 3; i++){
        for(j = i+1; j < 4; j++){
            if (a[i] == a[j]){
                return 0;
            }
        }
    }
    return 1;
}
```

Листинг А.1 – Исходный код игры, лист 1

```

int check(char a[]){//проверка на ввод числа
    int i, k = 0;
    if (strlen(a) != 4)
        return 1;
    for(i = 0; i < 4; i++){
        if (a[i] >= '0' && a[i] <= '9')
            k++;
    }
    if (k == 4) return 0;
    else return 1;
}

void input(int i, struct data *pl){ //ввод числа
    if (i != 0)
        printf ("Игрок %i\n", i);
    printf ("Введите число: ");
    if (pl->n < 9)
        scanf ("%s", &pl->number[pl->n]);
    else scanf ("%s", &pl->number[9]);
}

void print_table_one(struct data *pl){ // вывод поля для одного игрока
    int i, j = 1; //счетчики
    if (pl->n > 9){
        j = pl->n - 9;
    }
    system("cls");
    printf (" | Попытки\n");
    printf ("==|=====\\n");
    for(i = 0; (i < pl->n) && (i < 10); i++){
        printf("%2i| %s %iB %iK\\n", i+j, pl->number[i], pl->b[i], pl->k[i]);
    }
    printf("\\n");
}

//вывод поля для двух игроков
void print_table(struct data *pl_1, struct data *pl_2){
    int i;
    int j = 1;
    system("cls");
    printf (" | Игрок 1 | Игрок 2 | \\n");
    printf ("==|=====|=====|==\\n");
    if (pl_1->n > 9){
        j = pl_1->n - 9;
    }
    if (pl_1->n > pl_2->n){
        for(i = 0; (i < (pl_1->n - 1)) && (i < 9); i++){
            printf ("%2i| %s ", j+i, pl_1->number[i]);
            printf (" %iB %iK |", pl_1->b[i], pl_1->k[i]);
            printf (" %s %iB ", pl_2->number[i], pl_2->b[i]);
            printf ("%iK |%2i\\n", pl_2->k[i], i+j);
        }
        printf ("%2i| %s ", pl_1->n, pl_1->number[i]);
        printf (" %iB %iK |\\n", pl_1->b[i], pl_1->k[i]);
    }
    else if (pl_1->n == pl_2->n){
        for(i = 0; (i < pl_1->n) && (i < 10); i++){
            printf ("%2i| %s ", i+j, pl_1->number[i]);
            printf (" %iB %iK |", pl_1->b[i], pl_1->k[i]);
            printf (" %s %iB ", pl_2->number[i], pl_2->b[i]);
            printf ("%iK |%2i\\n", pl_2->k[i], i+j);
        }
    }
    printf("\\n");
}

```

Листинг А.1, лист 2

```

void save(struct data *pl_1, struct data *pl_2){
    FILE *fil;
    fil = fopen("game.txt", "w");
    fprintf(fil, "%i\n", pl_1->n);
    fprintf(fil, "%s\n", pl_1->guess);
    for (int i = 0; i < pl_1->n && i < 10; i++){
        fprintf(fil, "%s ", pl_1->number[i]);
    }
    fprintf(fil, "\n");
    fprintf(fil, "%i\n", pl_2->n);
    fprintf(fil, "%s\n", pl_2->guess);
    for (int i = 0; i < pl_2->n && i < 10; i++){
        fprintf(fil, "%s ", pl_2->number[i]);
    }
    fclose(fil);
}

void perenos(struct data *pl){
    for (int i = 0; i < 9; i++){
        for (int j = 0; j < 4; j++){
            pl->number[i][j] = pl->number[i+1][j];
        }
        pl->k[i] = pl->k[i+1];
        pl->b[i] = pl->b[i+1];
    }
}

int pr(int n, char *a){
    int i;
    for(i = 0; i < n; i++){
        if(a[i] == a[n])
            return 1;
    }
    return 0;
}

void old_game(struct data *pl_1, struct data *pl_2){
    FILE *f;
    f = fopen("game.txt", "r");
    int i;
    fscanf(f, "%d\n", &pl_1->n);
    fscanf(f, "%s\n", &pl_1->guess);
    for (i = 0; i < pl_1->n && i < 10; i++)
        fscanf(f, "%s", &pl_1->number[i]);
    fscanf(f, "%d\n", &pl_2->n);
    fscanf(f, "%s\n", &pl_2->guess);
    for (i = 0; i < pl_2->n && i < 10; i++)
        fscanf(f, "%s", &pl_2->number[i]);
    for (i = 0; i < pl_1->n && i < 10; i++){
        score_bk(0, i, pl_1, pl_2->guess);
    }
    for (i = 0; i < pl_2->n && i < 10; i++){
        score_bk(0, i, pl_2, pl_1->guess);
    }
    fclose(f);
}

```

Листинг А.1, лист 3


```

void game_one(){// одиночный режим
    struct data player;
    int i; // счетчик
    int v = 1; // выбор
    while (v){
        srand(time(NULL));
        for (i = 0; i < 4; i++){
            player.guess[i] = '0' + rand()%10;
            do{
                player.guess[i] = '0' + rand()%10;
            } while (pr(i, &player.guess[0]));
        }
        printf("\nКомпьютер загадал число. Попробуйте отгадать!\n");
        player.n = 0;
        player.w = 0;
        int g; // индекс последнего слова в массиве
        while (!player.w){
            g = player.n;
            if (player.n > 9){
                perenos(&player);
                g = 9;
            }
            input(0, &player);
            while(check(player.number[g]) || !(pr_num(&player.number[g][0]))){
                printf("Данные введены некорректно!\n");
                input(0, &player);
            }
            score_bk(1, player.n, &player, &player.guess[0]);
            print_table_one(&player); // сделать другой вывод поля
        }
        printf("Поздравляем! Вы выиграли!\n");
        printf("Количество ходов: %i \n\n", player.n);
        printf("Продолжить в этом режиме?\n1 - да, 0 - нет\n");
        scanf("%i", &v);
        while (v != 0 && v != 1){
            printf("Данные введены некорректно!\n");
            printf("Продолжить в этом режиме?\n1 - да, 0 - нет\n");
            fflush(stdin);
            scanf("%i", &v);
        }
    }
}

```

Листинг А.1, лист 4

```

void game_friend(int ex){ // с другом
    struct data player_1;
    struct data player_2;
    int i; //счетчик
    int v = 1; //выбор
    int g; // индекс последнего слова в массиве
    while (v){ //пока "да"
        player_1.n = 0;
        player_2.n = 0;
        player_1.w = 0;
        player_2.w = 0;
        if (ex){
            old_game(&player_1, &player_2);
            print_table(&player_1, &player_2);
            ex = 0;
            if (player_1.n > player_2.n){
                g = player_2.n;
                if (player_2.n > 9)
                    g = 9;
                input(2, &player_2);
                while(check(player_2.number[g]) || !(pr_num(&player_2.number[g][0]))){
                    printf("Данные введены некорректно!\n");
                    input(2, &player_2);
                }
                score_bk(0, player_2.n, &player_2, &player_1.guess[0]);
                player_2.n++;
                print_table(&player_1, &player_2);
                save(&player_1, &player_2);
            }
        }
        else{
            system("cls");
            printf("\n");
            printf("Игрок 1, введите загаданное число: ");
            scanf("%s", player_1.guess);
            while (check(player_1.guess) || !(pr_num(&player_1.guess[0]))){
                printf("Данные введены некорректно!\n");
                printf("Игрок 1, введите загаданное число: ");
                fflush(stdin);
                scanf("%s", player_1.guess);
            }
            system("cls");
            printf("\n");
            printf("Игрок 2, введите загаданное число: ");
            scanf("%s", player_2.guess);
            while (check(player_2.guess) || !(pr_num(&player_2.guess[0]))){
                printf("Данные введены некорректно!\n");
                printf("Игрок 2, введите загаданное число: ");
                fflush(stdin);
                scanf("%s", player_2.guess);
            }
            system("cls");
            printf("\n");
        }
    }
}

```

Листинг А.1, лист 5

```

while ((!player_1.w) && (!player_2.w)){
    g = player_1.n;
    if (player_1.n > 9){
        perenos(&player_1);
        perenos(&player_2);
        g = 9;
    }
    input(1, &player_1);
    while(check(player_1.number[g]) || !(pr_num(&player_1.number[g][0]))){
        printf("Данные введены некорректно!\n");
        input(1, &player_1);
    }
    score_bk(1, player_1.n, &player_1, &player_2.guess[0]);
    print_table(&player_1, &player_2);
    save(&player_1, &player_2);
    input(2, &player_2);
    while(check(player_2.number[g]) || !(pr_num(&player_2.number[g][0]))){
        printf("Данные введены некорректно!\n");
        input(2, &player_2);
    }
    score_bk(1, player_2.n, &player_2, &player_1.guess[0]);
    print_table(&player_1, &player_2);
    save(&player_1, &player_2);
}
if ((player_1.w) && (player_2.w)){
    printf("Ничья! \nКоличество ходов: %i\n\n", player_1.n);
    printf("Числа Игроков: ");
    for (i = 0; i < 4; i++){
        printf("%c", player_1.guess[i]);
    }
    printf("\n");
}
else if (player_1.w){
    printf("Игрок 1 выиграл! \nКоличество ходов: %i\n\n", player_1.n);
    printf("Число Игрока 1: ");
    for (i = 0; i < 4; i++){
        printf("%c", player_1.guess[i]);
    }
    printf("\n");
    printf("Число Игрока 2: ");
    for (i = 0; i < 4; i++){
        printf("%c", player_2.guess[i]);
    }
    printf("\n\n");
}
else if (player_2.w){
    printf("Игрок 2 выиграл! \nКоличество ходов: %i\n\n", player_2.n);
    printf("Число Игрока 1: ");
    for (i = 0; i < 4; i++){
        printf("%c", player_1.guess[i]);
    }
    printf("\n");
    printf("Число Игрока 2: ");
    for (i = 0; i < 4; i++){
        printf("%c", player_2.guess[i]);
    }
    printf("\n\n");
}
}

```

Листинг А.1, лист 6

```

        remove("game.txt");
        printf("Продолжить в этом режиме?\n1 - да, 0 - нет\n");
        scanf("%i", &v);
        while (v != 0 && v != 1){
            printf("Продолжить в этом режиме?\n1 - да, 0 - нет\n");
            fflush(stdin);
            scanf("%i", &v);
        }
        player_1.n = 0;
        player_2.n = 0;
    }
}

void main(){
    int i; // счетчик
    int r, v = 1; // r - режим
    printf("Добрый день! Это игра \"Быки и коровы!\"\n");
    printf("Правила игры:\n");
    printf("игроки загадывают четырехзначное число без повтора цифр.\nПотом они
задают друг другу другие такие четырехзначные числа, какие на их взгляд у
противника.\nЕсли совпало место и число, то это бык (Б). Если совпало только
число, то это корова (К).\n");
    FILE *f;
    f = fopen("game.txt", "r");
    while (v == 1){
        if (f){
            fclose(f);
            printf("\nВыберете режим:\n0 - Продолжить\n1 - одиночный\n2 - с другом\n");
            scanf("%i", &r);
            while (r != 1 && r != 2 && r != 0){
                printf("Данные введены некорректно!\n");
                printf("Выберете режим:\n0 - Продолжить\n1 - одиночный");
                printf("\n2 - с другом\n");
                fflush(stdin);
                scanf("%i", &r);
            }
        }
        else{
            printf("\nВыберете режим:\n1 - одиночный\n2 - с другом\n");
            scanf("%i", &r);
            while (r != 1 && r != 2){
                printf("Данные введены некорректно!\n");
                printf("Выберете режим:\n1 - одиночный\n2 - с другом\n");
                fflush(stdin);
                scanf("%i", &r);
            }
        }
        if (r == 1){
            game_one();
        }
        else if (r == 2){
            game_friend(0);
        }
        else if (r == 0){
            game_friend(1);
        }
        printf("Желаете сыграть еще?\n1 - да, 0 - нет\n");
        scanf("%i", &v);
        while (v != 0 && v != 1){
            printf("Данные введены некорректно!\n");
            printf("Желаете сыграть еще?\n1 - да, 0 - нет\n");
            fflush(stdin);
            scanf("%i", &v);
        }
    }
}

```

Приложение Б
(обязательное)
Пример работы программы

Начинаем новую и выбираем режим 2 (рисунок 11).

```
G:\Курсовая\Код>a
Добрый день! Это игра "Быки и коровы!"
Правила игры:
игроки загадывают четырехзначное число без повтора цифр.
Потом они задают друг другу другие такие четырехзначные числа, какие на их взгляд у противника.
Если совпало место и число, то это бык (Б). Если совпало только число, то это корова (К).

Выберете режим:
1 - одиночный
2 - с другом
2
```

Рисунок 11 – Начало игры

Вводим загаданные числа, для каждого из игроков (рисунки 12 и 13).

Игрок 1, введите загаданное число: 2586

Рисунок 12 – Введение числа первым игроком

Игрок 2, введите загаданное число: 2369

Рисунок 13 – Введение числа вторым игроком

На рисунке 14 изображены попытки игроков.

	Игрок 1	Игрок 2	
1	3689 1Б 2К	2589 3Б 0К	1
2	9632 0Б 4К	1028 0Б 2К	2
3	7456 0Б 1К	9832 0Б 2К	3
4	0215 0Б 1К		

Игрок 2
Введите число:

Рисунок 14 – Процесс игры

Игрок 1 отгадывает число Игрока 1 (рисунок 15).

	Игрок 1	Игрок 2	
1	3689 1Б 2К	2589 3Б 0К	1
2	9632 0Б 4К	1028 0Б 2К	2
3	7456 0Б 1К	9832 0Б 2К	3
4	0215 0Б 1К	2583 3Б 0К	4
5	2369 4Б 0К	2584 3Б 0К	5

Игрок 1 выиграл!
Количество ходов: 5

Число Игрока 1: 2586
Число Игрока 2: 2369

Продолжить в этом режиме?
1 - да, 0 - нет
1

Рисунок 15 – Окончание игры выигрышем первого игрока
Выходим из режима и игры (рисунок 16).

Продолжить в этом режиме?

1 - да, 0 - нет

0

Желаете сыграть еще?

1 - да, 0 - нет

0

G:\Курсовая\Код>

Рисунок 16 – Выход из игры