**МИНОБРНАУКИ РОССИИ**

**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**

**ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**

**Кафедра САПР**

**ОТЧЁТ**

**По курсовой работе №1**

**По дисциплине «Программирование»**

Студент гр. 3351 \_\_\_\_\_\_\_\_\_\_\_\_ Морозов А.А.

Преподаватель \_\_\_\_\_\_\_\_\_\_\_\_ Рыжов Н.Г.

Санкт-Петербург

2024

цель работы

Игра быки и коровы. Игрок1 загадывает четырехзначное число, состоящее из различных цифр. Игрок2 отгадывает задуманное Игроком1 число, перебирая возможные варианты. Каждый раз Игрок2 предлагает возможный вариант числа, а Игрок1 делает Игроку2 подсказку: сообщает количество быков и коров, после чего Игрок2 с учетом подсказки продолжает отгадывание числа до тех пор, пока не отгадает. Быки – это количество цифр в предложенном Игроком2 числе, совпадающих по значению и стоящих в правильной позиции в задуманном Игроком1 числе. Коровы – количество цифр, совпадающих по значению, но находящихся в неверной позиции. Например, если Игрок1 задумал число 5671, а Игрок2 предложил вариант 7251, то число быков равно 1 (только цифра 1 стоит на своем месте), а число коров равно 2 (цифры 7 и 5 присутствуют в загаданном числе, но стоят не на своих местах). Составить программу для игры человека с компьютером. Вариант курсовой работы – 14.

**Текст программы Быки и коровы**

#include <iostream>

#include <ctime>

#include <vector>

#include <algorithm>

#include <sstream>

using namespace std;

// Функция генерации случайного четырёхзначного числа, в котором цифры не повторяются

int generate\_random\_number() {

vector<int> digits{ 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 };

int random\_number = 0;

for (size\_t i = 0; i < 4; i++) {

size\_t index = rand() % (digits.size() - i);

random\_number = random\_number \* 10 + digits[index];

digits[index] = digits[digits.size() - 1 - i];

}

return random\_number;

}

// Функция вывода правил игры

void rules() {

cout << "Правила игры `Быки и коровы`:" << endl;

cout << "Компьютер загадывает четырехзначное число, состоящее из различных цифр. Игрок отгадывает задуманное компьютером число, перебирая возможные варианты. Каждый раз Игрок предлагает возможный вариант числа, а компьютер делает Игроку подсказку: сообщает количество быков и коров, после чего Игрок с учетом подсказки продолжает отгадывание числа до тех пор, пока не отгадает. Быки – это количество цифр в предложенном Игроком2 числе, совпадающих по значению и стоящих в правильной позиции в задуманном Игроком1 числе. Коровы – количество цифр, совпадающих по значению, но находящихся в неверной позиции. Например, если компьютер задумал число 5671, а Игрок предложил вариант 7251, то число быков равно 1 (только цифра 1 стоит на своем месте), а число коров равно 2 (цифры 7 и 5 присутствуют в загаданном числе, но стоят не на своих местах).\n" << endl;

}

// Правильное ли число

bool checkNum(string number) {

if (number.length() != 4) {

return false;

}

return true;

}

// Функция игры

void game() {

srand(static\_cast<unsigned int>(time(0))); // Установка рандомайзера по времени

int data; // Переменная для хранения числа сгенерированного компьютером

string number; // Строка для хранения чисел пользователя

bool run = true; // Булл для цикла

int attempts = 0; // Переменная для хранения количества попыток

int bull = 0, cow = 0; // Создание подсчёта количества быков и количества коров

cout << "Компьютер загадал четырёхзначное число: \*\*\*\*." << endl;

data = generate\_random\_number(); // Генерация рандомного числа и запись в data

while (run) { // Цикл хода игры

cout << "Введите своё число: ";

cin >> number; // Ввод пользователем числа

if (number == to\_string(0)) { // Сдаться при вводе 0

cout << "Вы сдались! Возвращение в меню..." << endl;

cout << "Компьютер загадывал число " << data << ".\n" << endl;

return;

}

while (!checkNum(number)) { // Цикл-проверка, что число четырёхзначное

cout << "Вы ввели некорректное число. Повторите ввод: ";

cin >> number; // Повторный ввод числа

}

attempts++; // Увеличение счётчика попыток

if (number != to\_string(data)) { // Игрок угадал число?

for (int i = 0; i < number.length(); i++) { // Цикл для подсчёта быков

if (0 >= number[i] <= 9 and 0 >= to\_string(data)[i] <= 9 and number[i] == to\_string(data)[i]) {

bull++; // Увеличение счётчика быков

}

}

for (int i = 0; i < number.length(); i++) { // Цикл для подсчёта коров

for (int j = 0; j < to\_string(data).length(); j++) {

if (0 >= number[i] <= 9 and 0 >= to\_string(data)[j] <= 9 and number[i] == to\_string(data)[j] and number[i] != to\_string(data)[i]) {

cow++; // Увеличение количества коров

}

}

}

if (bull != 0) { // Вывод быков

cout << bull << "Б";

}

if (cow != 0) { // Вывод коров

cout << cow << "К";

}

if (bull == 0 and cow == 0) { // Если быков и коров нет

cout << "Нет";

}

cout << endl;

cow = 0; // Обнуление счётчиков для следующего раунда

bull = 0;

}

else {

cout << "Вы выиграли. Компьютер загадал число " << data << ".\n";

cout << "Количество ходов: " << attempts << ".\n" << endl;

return;

}

}

}

// Эта функция проверяет, является ли число допустимым ответом

bool isValidGuess(int num) {

vector<int> digits; // Хранение цифр числа здесь

while (num > 0) {

int digit = num % 10; // Извлеките последнюю цифру числа

if (find(digits.begin(), digits.end(), digit) != digits.end()) {

return false; // Если цифра уже находится в векторе, число недопустимо

}

digits.push\_back(digit); // Добавить цифру в вектор

num /= 10; // Удалить последнюю цифру числа

}

return true; // Если число содержит 4 разные цифры, оно является допустимым ответом

}

// Эта функция подсчитывает количество "быков"

int countBulls(int guess, int data) {

int bulls = 0;

for (int i = 0; i < 4; ++i) { // Перебор 4 цифры чисел

if ((guess % 10) == (data % 10)) { // Проверка, является ли последняя цифра ответа и данных одинаковой

bulls++;

}

guess /= 10; // Удалить последнюю цифру ответа

data /= 10; // Удалить последнюю цифру данных

}

return bulls;

}

// Эта функция подсчитывает количество "коров"

int countCows(int guess, int data) {

int cows = 0;

vector<int> guessDigits(4), dataDigits(4); // Хранение цифр ответа и данных здесь

for (int i = 0; i < 4; ++i) { // Переберите 4 цифры чисел

guessDigits[3 - i] = guess % 10; // Извлечь последнюю цифру ответа и сохранить ее в векторе

dataDigits[3 - i] = data % 10; // Извлечь последнюю цифру данных и сохранить ее в векторе

guess /= 10; // Удалить последнюю цифру ответа

data /= 10; // Удалить последнюю цифру данных

}

for (int i = 0; i < 4; ++i) { // Переберите 4 цифры ответа

if (std::find(dataDigits.begin(), dataDigits.end(), guessDigits[i]) != dataDigits.end() // Проверьте, находится ли цифра в данном

and guessDigits[i] != dataDigits[i]) {

cows++;

}

}

return cows;

}

// Основная функция теста компьютера

void computerGuess(int data) {

int guess; // Ответ компьютера

int attempts = 0;

vector<int> possibleNums; // Вектор возможных чисел

for (int i = 1023; i <= 9876; ++i) {

if (isValidGuess(i)) { // Проверить, является ли число допустимым ответом

possibleNums.push\_back(i);

}

}

while (!possibleNums.empty()) { // Цикл, пока компьютер не угадает данные

guess = possibleNums.front(); // Компьютер угадывает первое число в векторе

attempts++;

cout << "Компьютер ввёл число: " << guess << endl;

int bulls = countBulls(guess, data); // Подсчёт количества "быков" в ответе

int cows = countCows(guess, data); // Подсчёт количества "коров" в ответе

if (bulls != 0) {

cout << bulls << "Б";

}

if (cows != 0) {

cout << cows << "К";

}

if (bulls == 0 and cows == 0) {

cout << "Нет";

}

cout << endl;

if (bulls == 4) { // Если компьютер угадал данные

cout << "Компьютер угадал число: " << guess << endl;

cout << "Количество попыток: " << attempts << endl;

return;

}

// Удалить из вектора числа, которые не являются возможными ответами на основе количества "быков" и "коров"

possibleNums.erase(remove\_if(possibleNums.begin(), possibleNums.end(),

[&](int x) { return countBulls(x, guess) != bulls or countCows(x, guess) != cows; }),

possibleNums.end());

}

}

int main() {

setlocale(LC\_ALL, "RU");

int choice;

// Запуск главного цикла программы

while (true) {

cout << "-=-=- Меню игры `Быки и коровы` -=-=-" << endl;

cout << "1. Правила" << endl;

cout << "2. Начать игру" << endl;

cout << "3. Тест компьютера" << endl;

cout << "4. Выйти\n" << endl;

cin >> choice; // Ввод выбора

cout << endl;

if (choice == 1) { // Правила игры

rules();

}

else if (choice == 2) { // Собственно сама игра

game();

}

else if (choice == 3) { // Тест-игра компьютера

int data;

cout << "Загадайте четырёхзначное число (цифры не должны повторяться): ";

cin >> data;

while (!checkNum(to\_string(data)) or !isValidGuess(data)) {

cout << "Вы ввели некорректное число. Повторите ввод: ";

cin >> data; // Повторный ввод числа

}

computerGuess(data);

}

else if (choice == 4) { // Выход из программы

return 0;

}

else { // Неверный ввод

cout << "Неверный ввод. Перезапуск меню.\n" << endl;

}

}

}

**Описание программы Быки и коровы**

Программное обеспечение – Microsoft Visual Studio 2022. Язык программирования – C++. При запуске выводится меню. В нём можно узнать правила, начать игру с угадыванием числа, начать игру с загадыванием числа, а также завершить работу программы. В любой момент работы программы её работу можно остановить или вернуться назад. После каждой игры на экран снова выводится меню.

**Руководство пользователя к программе Быки и коровы**

При запуске программа выводит меню с четырьмя пунктами. Оно может быть доступно в любой момент работы программы, а также выводится после завершения каждой игры. (Рис. 1)

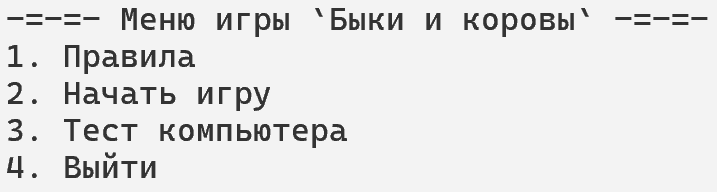


Рис. 1 – Меню программы

Первый пункт выводит правила игры Быки и коровы для последующего ознакомления пользователя. После вывода правил снова открывается главное меню. (Рис. 2)

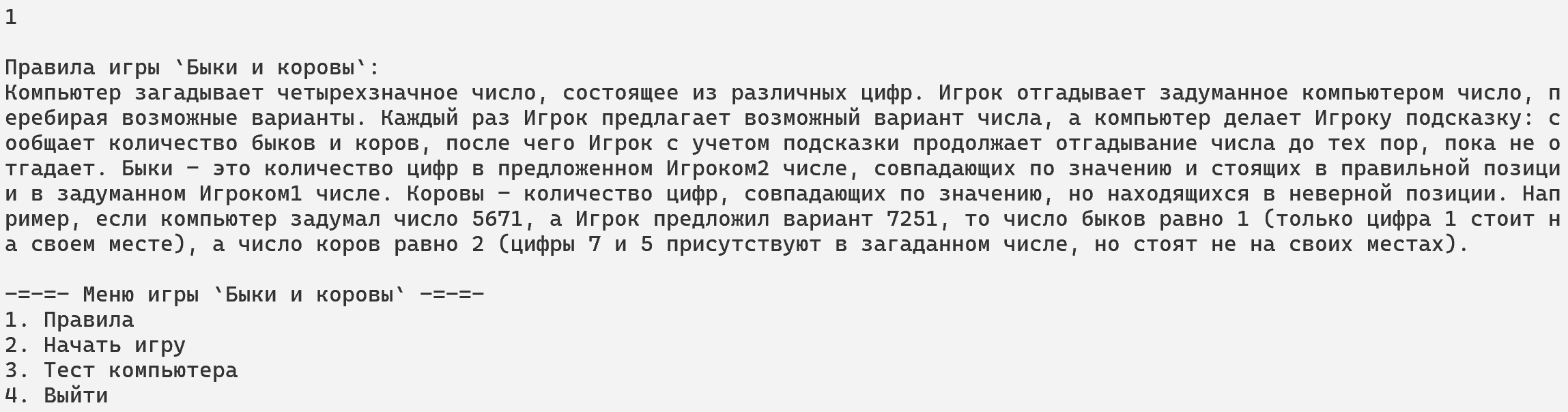


Рис. 2 – Пункт Правила

Во втором пункте пользователь может начать игру, где компьютер загадает четырёхзначное число, цифры в котором не повторяются. Пользователь должен будет отгадывать загаданное число ориентируясь по подсказкам компьютера, числу быков и коров. Для работы необходимо лишь вводить варианты чисел. (Рис. 3, Рис. 5)

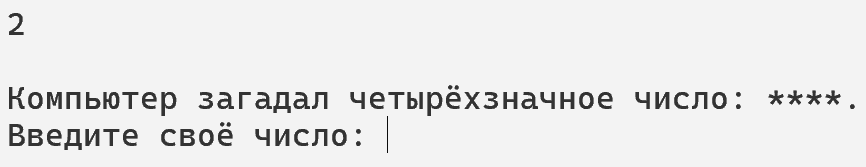


Рис. 3 – Ввод числа

Если число вводится неправильно, то его необходимо будет ввести повторно некоторое количество раз, пока оно не будет введено верно. (Рис. 4)

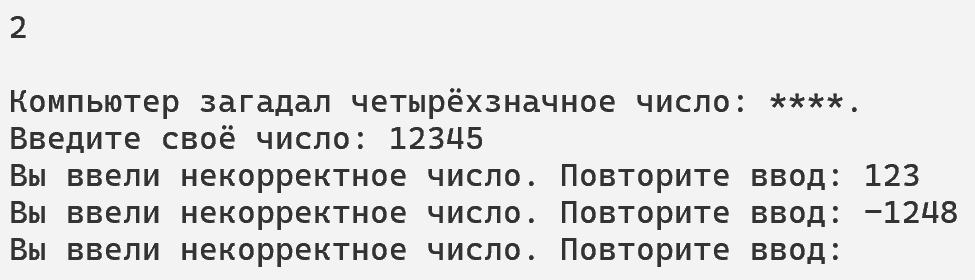


Рис. 4 – Некорректный ввод

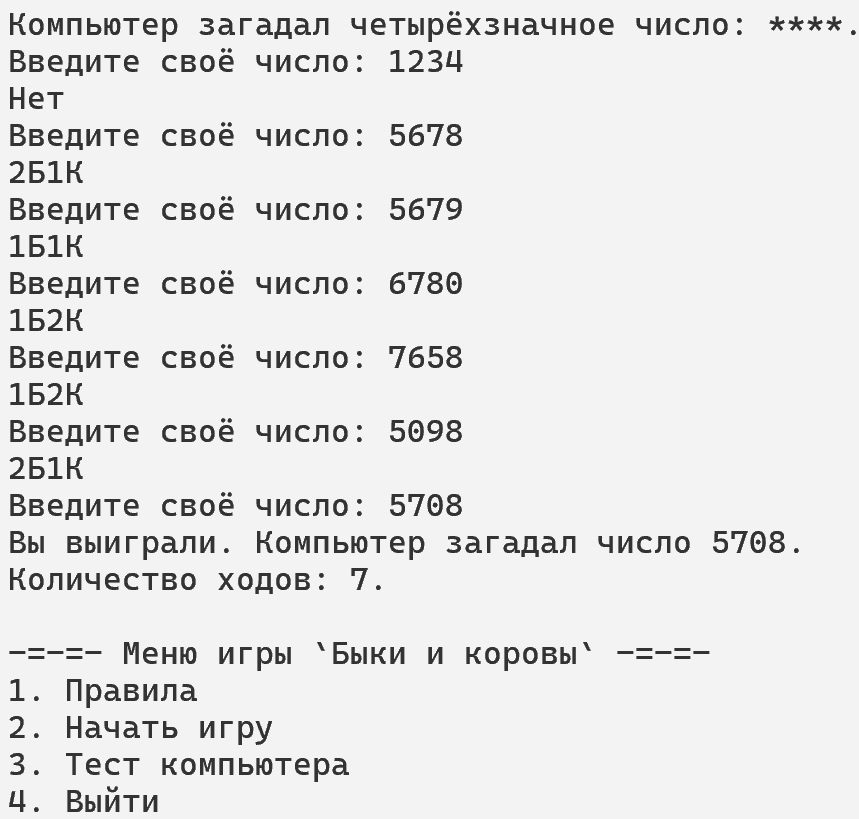


Рис. 5 – Работа программы

Также в третьем пункте меню пользователь может загадать число, чтобы его угадывал компьютер. Он делает это в автоматическом режиме, то есть пользователь не сам указывает количество быков и коров в предложенном компьютером варианте. Это сделано во избежание ошибок в работе программы на основе человеческого фактора. (Рис. 6, Рис. 7)



Рис. 6 – Загадать число



Рис. 7 – Работа программы

**Пути дальнейшего улучшения программы**

1. Улучшение визуальной части программы
2. Модернизация интеллекта компьютера для отгадывания числа