// Ex.1.1

#include <iostream>

using namespace std;

int main()

{

setlocale(0, "RUS");

cout << " Игра \"Угадай число\"\n" << "\n";

cout << " Загадай число в диапазоне от 1 до 1000. Я попробую угадать его!\n" << "\n";

cout << " Введите загаданное число:(для проверки)\n"; // checking number

int z; // checking number

cin >> z; // checking number

cout << " Меню выбора операций:\n" << "\n";

cout << " 1 - да, ты угадал!\n";

cout << " 2 - нет, моё число меньше!\n";

cout << " 3 - нет, моё число больше!\n";

cout << " 0 - выход из программы.\n" << "\n";

int choice = 5; // выбор операции

int trys = 0; // кол-во попыток

int left = 0, right = 1000, mid = 0; // переменные границ диапазона и середины

do

{

mid = (left + right) / 2; // находим среднее значение (искомое число)

cout << " Твоё число " << mid << ". Я угадал?\n";

cin >> choice;

trys++;

while (choice != 0)

{

if (choice == 1)

{

cout << " Урааа! Я угадал.\n" << " Использовано попыток - " << trys << "\n";

system("color A0");

break;

}

if (choice == 0)

{

cout << " Пока! До новых встреч!\n";

system("color 90");

break;

}

if (choice == 2)

{

right = mid; // изменение значения правой границы

mid = (left + right) / 2; // находим среднее значение (искомое число)

cout << " Твоё число " << mid << ". Я угадал?\n";

cin >> choice;

if (left + 2 == right && ((choice == 2 || choice == 3))) break; // случай обмана программы

trys++;

}

if (choice == 3)

{

left = mid; // изменение значения правой границы

mid = (left + right) / 2; // находим среднее значение (искомое число)

cout << " Твоё число " << mid << ". Я угадал?\n";

cin >> choice;

if (left + 2 == right && ((choice == 2 || choice == 3))) break; // случай обмана программы

trys++;

}

}

if (left + 2 == right && ((choice == 2 || choice == 3))) // случай обмана программы

{

cout << " Вы обманули меня!!!\n";

system("color 40");

break;

}

if (choice == 0)

{

cout << " Пока! До новых встреч!\n";

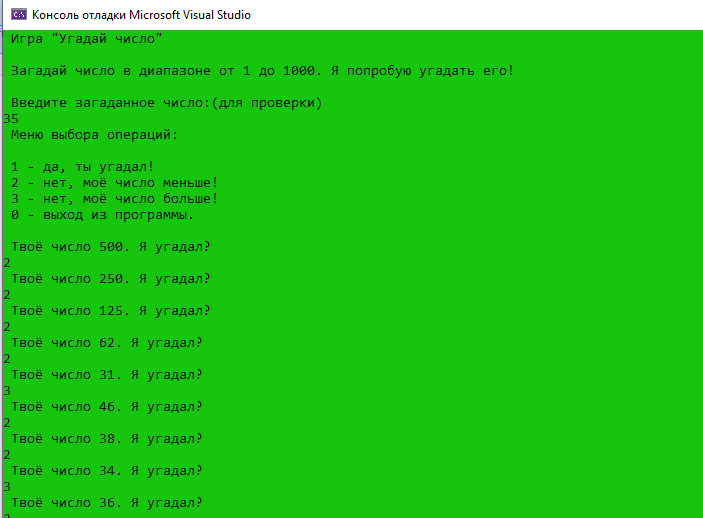
system("color 90");

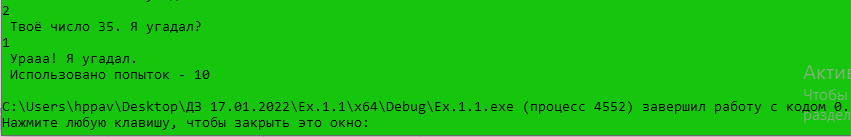
break;

}

} while (choice != 1);

}





// Ex.1.2

#include <iostream>

using namespace std;

int main()

{

setlocale(0, "RUS");

cout << " Количество и номера всех счастливых трамвайных билетов с шестизначными номерами.\n" << "\n";

int number = 1;// начало диапазона

int counter = 0; // кол-во счастливых билетов

int a1, a2, a3, a4, a5, a6; // каждая цифра числа

int b, c; // сумма первых чисел, сумма последних чисел

do

{

a1 = number % 10; // 6 число

a2 = number % 100 / 10; // 5 число

a3 = number % 1000 / 100; // 4 число

a4 = number % 10000 / 1000; // 3 число

a5 = number % 100000 / 10000; // 2 число

a6 = number / 100000; // 1 число

b = a6 + a5 + a4;

c = a3 + a2 + a1;

if (b == c)

{

cout << a6 << a5 << a4 << a3 << a2 << a1 << " ";

counter++;

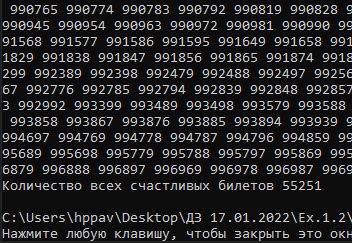
}

number++;

} while (number > 0 && number <= 999999);

cout << " \nКоличество всех счастливых билетов " << counter << "\n";

}



// Ex.1.3

#include <iostream>

using namespace std;

int main()

{

setlocale(0, "RUS");

cout << " Показать на экран все восьмизначные числа и их кол-во,\n";

cout << " цифры в которых не повторяются и делятся без остатка на 12345.\n" << "\n";

int n = 10000000; // начало диапазона

int a1, a2, a3, a4, a5, a6, a7, a8; // каждая цифра числа

int counter = 0; // кол-во найденых чисел

do

{

a1 = n % 10; // 8 число

a2 = n % 100 / 10; // 7 число

a3 = n % 1000 / 100; // 6 число

a4 = n % 10000 / 1000; // 5 число

a5 = n % 100000 / 10000; // 4 число

a6 = n % 1000000 / 100000; // 3 число

a7 = n % 10000000 / 1000000; // 2 число

a8 = n / 10000000; // 1 число

if (a1 != a2 && a1 != a3 && a1 != a4 && a1 != a5 && a1 != a6 && a1 != a7 && a1 != a8 &&

a2 != a3 && a2 != a4 && a2 != a5 && a2 != a6 && a2 != a7 && a2 != a8 &&

a3 != a4 && a3 != a5 && a3 != a6 && a3 != a7 && a3 != a8 &&

a4 != a5 && a4 != a6 && a4 != a7 && a4 != a8 &&

a5 != a6 && a5 != a7 && a5 != a8 &&

a6 != a7 && a6 != a8 &&

a7 != a8 &&

n % 12345 == 0)

{

cout << n << " ";

counter++;

}

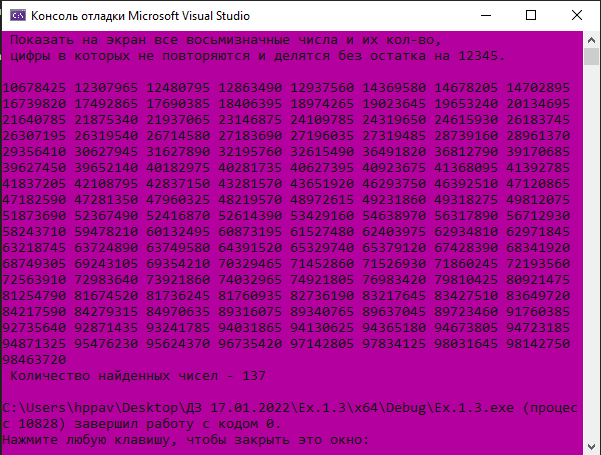
n++;

} while (n >= 10000000 && n <= 99999999);

cout << "\n Количество найденных чисел - " << counter << "\n";

system("color D0");

}



// Ex.1.4

#include <iostream>

#include <ctime>

#include <string>

#include <windows.h>

#include <fstream>

#include <string>

using namespace std;

int main()

{

setlocale(0, "RUS");

cout << " Игра \"Однорукий бандит\"\n" << "\n";

cout << " Правила игры\n" << "\n";

string getcontent1;

ifstream openfile1("rules.txt");

if (openfile1.is\_open())

{

while (getline(openfile1, getcontent1))

{

cout << getcontent1 << "\n" << "\n";

}

openfile1.close();

}

else

{

cout << " Файл не найден!\n";

}

const int removal = 1000; // снятие за каждый показ комбинации

int credit = 5000; // начальный счёт игрока

int n1 = 0;

int n2 = 0;

int n3 = 0; // рандомные числа

int choice;// выбор операции

do

{

cout << " На вашем счету " << credit << " кредитов\n";

cout << " Меню операций:\n";

cout << " Крутить барабан - 1\n";

cout << " Остановить игру - 0\n";

cin >> choice;

srand(time(0)); // автоматическая рандомизация

n1 = 1 + rand() % 7;

n2 = 1 + rand() % 7;

n3 = 1 + rand() % 7;

if (choice == 1)

{

string str = " " + to\_string(n1) + " " + to\_string(n2) + " " + to\_string(n3);// конвертация int в строку

MessageBoxA(NULL, str.c\_str(), " Комбинация", MB\_OK); // вывод комбинации в окно

//cout << n1 << n2 << n3 << "\n" << "\n";

credit -= removal;

}

if (choice == 1 && (n1 == 0 && n2 == 0 && n3 == 0 ||

n1 == 1 && n2 == 1 && n3 == 1 ||

n1 == 2 && n2 == 2 && n3 == 2 ||

n1 == 3 && n2 == 3 && n3 == 3 ||

n1 == 4 && n2 == 4 && n3 == 4 ||

n1 == 5 && n2 == 5 && n3 == 5))

{

credit += 3000;

system("color 20");

Sleep(500);

system("color 0F");

cout << " Призовая комбинация!\n" << " Вы выйграли 3000 кредитов.\n" << "\n";

}

else if (choice == 1 && (n1 == 5 || n1 == 5 && n2 == 5)) // призовая комбинация

{

credit += 2000;

system("color 20");

Sleep(500);

system("color 0F");

cout << " Призовая комбинация!\n" << " Вы выйграли 2000 кредитов.\n" << "\n";

}

if (choice == 1 && (n1 == 6 && n2 == 6 && n3 == 6)) // штрафная комбинация

{

credit -= 3000;

system("color C0");

Sleep(500);

system("color 0F");

cout << " Штрафная комбинация!\n" << " Вы потеряли 3000 кредитов.\n" << "\n";

}

if (choice == 1 && (n1 == 7 && n2 == 7 && n3 == 7))// комбинация джек-пота

{

credit += 5000;

system("color 20");

Sleep(500);

system("color 10");

Sleep(500);

system("color 50");

Sleep(500);

system("color 0F");

cout << " Джек-пот!!!\n" << " Вы выйграли!\n";

cout << " На вашем счету " << credit << " кредитов.\n";

MessageBoxA(NULL, " Спасибо за игру. Заберите свой выигрышь в кассе!", " ", MB\_OK); // конец игры (выигрышь)

//cout << " Спасибо за игру. Заберите свой выигрышь в кассе.\n";

break;

}

else if (choice == 1 && (n1 == 7 || n1 == 7 && n2 == 7)) // призовая комбинация

{

credit += 2000;

system("color 20");

Sleep(500);

system("color 0F");

cout << " Призовая комбинация!\n" << " Вы выйграли 2000 кредитов.\n" << "\n";

}

if (choice == 0)

{

credit = 0;

cout << " Вы завершили игру\n";

break;

}

} while (credit != 0);

if (credit == 0)

{

system("color 40");

Sleep(500);

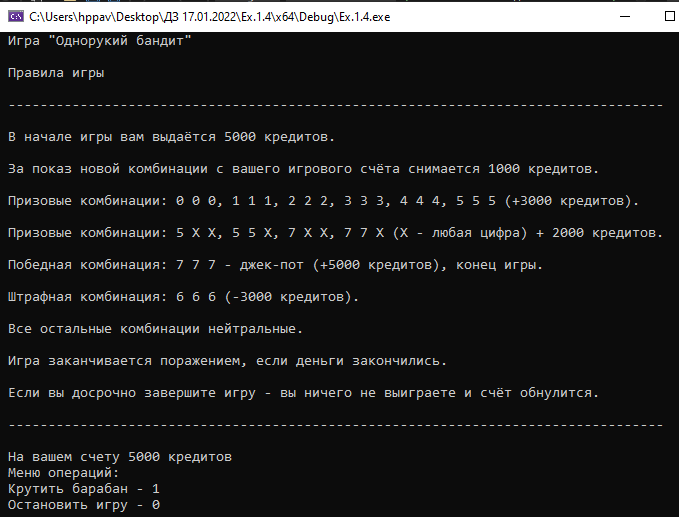
system("color 0F");

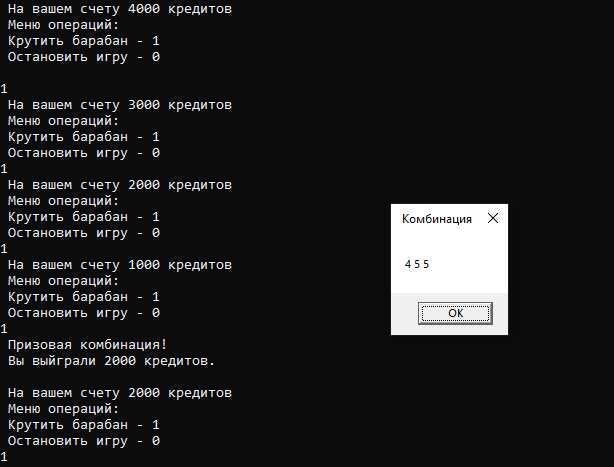
cout << " На вашем счету " << credit << " кредитов\n";

cout << " Попытайте удачу в другой раз!\n";

}

}





// Ex.1.5

#include <iostream>

using namespace std;

int main()

{

setlocale(0, "RUS");

cout << " Рассчитать минимальное количество копейщиков,\n";

cout << " которое необходимо, чтобы убить дракона.\n" << "\n";

cout << " Введите данные очков здоровья дракона:\n";

int hp\_dragon;

cin >> hp\_dragon;

cout << " Введите данные очков атаки дракона:\n";

int atack\_dragon;

cin >> atack\_dragon;

cout << " Введите данные очков здоровья 1-го копейщика:\n";

int hp\_pike;

cin >> hp\_pike;

cout << " Введите данные очков атаки 1-го копейщика:\n";

int atack\_pike;

cin >> atack\_pike;

int hp\_dragon2 = hp\_dragon;

int atack\_dragon2 = atack\_dragon;

int hp\_pike2 = hp\_pike;

int atack\_pike2 = atack\_pike;

int x = 1; // кол-во копейщиков

int hp\_allpike = x \* hp\_pike;

int atack\_allpike = x \* atack\_pike;

int hp\_dragon1 = hp\_dragon;

int x1 = 0;

cout << " Рассчёт циклом нужного кол-ва копейщиков для победы\n" << "\n";

while (hp\_dragon > 0)

{

cout << " Атакуют копейщики:\n";

hp\_dragon -= atack\_allpike;

cout << " Осталось жизни у дракона " << hp\_dragon << "\n" << "\n";

if (hp\_dragon <= 0) break;

cout << " Атакует дракон:\n";

hp\_allpike -= atack\_dragon;

cout << " Осталось жизни у копейщиков " << hp\_allpike << "\n" << "\n";

if (hp\_allpike <= 0)

{

if (x < x1)

{

x = x1;

}

x++;

hp\_dragon = hp\_dragon1;

x1 = x;

hp\_allpike = x \* hp\_pike;

atack\_allpike = x \* atack\_pike;

}

if (hp\_allpike > 0)

{

x = hp\_allpike / hp\_pike;

if (hp\_allpike % hp\_pike != 0)

{

x = x + 1;

hp\_allpike = x \* hp\_pike - (hp\_pike - hp\_allpike % hp\_pike);

atack\_allpike = x \* atack\_pike;

}

if (hp\_allpike % hp\_pike == 0)

{

hp\_allpike = x \* hp\_pike;

atack\_allpike = x \* atack\_pike;

}

}

}

cout << x1 << " - минимальное кол-во копейщиков для победы над драконом\n";

cout << hp\_dragon2 << " - хп дракона\n";

cout << atack\_dragon2 << " - урон дракона\n";

cout << hp\_pike2 << " - хп копейщика\n";

cout << atack\_pike2 << " - урон копейщика\n";

cout << " ЛОГ БОЯ \n" << "\n";

int round = 1;

int hp\_allpike2 = x1 \* hp\_pike2;

int atack\_allpike2 = x1 \* atack\_pike2;

do

{

cout << " РАУНД " << round << "\n" << "\n";

cout << " Атакуют копейщики:\n";

hp\_dragon2 -= atack\_allpike2;

if (hp\_dragon2 <= 0)

{

hp\_dragon2 = 0;

cout << " Копейщики победили!!! Дракон курит трубку!!!\n";

system("color 20");

break;

}

cout << " Осталось жизни у дракона " << hp\_dragon2 << "\n" << "\n";

cout << " Атакует дракон:\n";

hp\_allpike2 -= atack\_dragon2;

cout << " Осталось жизни у копейщиков " << hp\_allpike2 << "\n" ;

x1 = hp\_allpike2 / hp\_pike2;

if (hp\_allpike2 % hp\_pike2 == 0)

{

cout << " Осталось копейщиков " << x1 << "\n" << "\n";

hp\_allpike2 = x1 \* hp\_pike2;

atack\_allpike2 = x1 \* atack\_pike2;

}

if (hp\_allpike2 % hp\_pike2 != 0)

{

x1 += 1;

cout << " Осталось копейщиков " << x1 << ", из которых 1 ранен, у него осталось " << hp\_allpike2 % hp\_pike2 << " очков здоровья\n" << "\n";

hp\_allpike2 = x1 \* hp\_pike2 - (hp\_pike2 - hp\_allpike2 % hp\_pike2);

atack\_allpike2 = x1 \* atack\_pike2;

}

round++;

} while (!(hp\_dragon2 <= 0));

}

// вариант с убывающей прогрессией только для целых чисел кратным 10

//double a = (atack\_dragon / hp\_pike) \* atack\_pike; // на сколько уменьшается наносимый урон копейщиками за 1 атаку

//cout << a << "\n";

//int S = ceil(hp\_dragon / a); // количество атак которые нужно нанести копейщиками которые умирают за одну атаку дракона,

// // это и есть суммой членов прогрессии , первым числом которой явлеться -1-, с у четом того что копейщики бьют первыми

//cout << S << "\n";

//int x = ceil(abs((1 - sqrt(1 + (8 \* S))) / 2)); //оптимальное корень квадратного уравнения,

// //которое выводиться из формулы суммы арифметической прогрессии.

//cout << x << "\n";

//double m = x \* a; // необходимый урон копейщиками в первой атаке для того чтобы убить дракона.

//cout << m << "\n";

//int reccomended\_quantity = ceil(m / atack\_pike); // кол-во копейщиков для победы

//

//cout << reccomended\_quantity;

