

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«ПЕРМСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

ЭЛЕКТРОТЕХНИЧЕСКИЙ ФАКУЛЬТЕТ

КАФЕДРА «ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И
АВТОМАТИЗИРОВАННЫХ СИСТЕМ»

ОТЧЁТ ПО ИГРЕ
«ОТГАДАЙ СЛУЧАЙНОЕ ЧИСЛО»

Дисциплина: «Основы алгоритмизации и программирования»

Выполнил:

Студент группы ИВТ-21-26
Безух Владимир Сергеевич

Проверил:

Доцент кафедры ИТАС
Полякова Ольга Андреевна

Пермь,
Октябрь 2021

Постановка задачи

Реализовать программу «Отгадай случайное число» (далее — игра). Пользователю (далее — игроку) предлагается угадать число в заранее заданном диапазоне целых чисел за отведённое число попыток. После каждой попытки игрок получает информацию о том, является ли введённое число большим или меньшим по отношению к загаданному игрой числу.

В качестве демонстрации работы игры выбран диапазон от 1 до 100 с 7-ю попытками. Эти параметры привязаны к константам, которые можно легко изменить при необходимости. Частично или целиком диапазон может содержать в себе отрицательные целые числа.

Общий анализ задачи

Для генерации псевдослучайных чисел используется версия алгоритма «Вихрь Мерсенна» под названием «MT19937», период последовательности которого составляет $2^{19937} - 1$. Это наиболее оптимальный вариант, предлагаемый стандартной библиотекой C++, с точки зрения источника энтропии и производительности.

Равномерное распределение полученных значений по диапазону [MIN, MAX] можно осуществить по следующей формуле `mersenne() % (MAX - MIN + 1) + MIN`. Однако наиболее оптимальным вариантом является использование уже готового решения в виде `uniform_int_distribution<> distribution(MIN, MAX)`.

Игра осуществляется до тех пор, пока игрок не угадает число или не исчерпает заранее известное число попыток. Сложность игры можно корректировать, изменяя количество попыток. Ориентироваться в этом случае нужно на метод поиска «разделяй и властвуй» или, проще говоря, на бинарный поиск, позволяющий найти решение игры за не более чем $\log_2(N)$ шагов, где N — количество чисел в диапазоне.

После окончания игры игроку предлагается повторить игровую сессию или завершить исполнение программы.

Демонстрация исполнения кода

В рамках демонстрации работы кода его участки расположены в удобной для восприятия последовательности. Расположение этих участков в исходном коде может отличаться в соответствии со всеми особенностями языка программирования C++.

```
#include <random>
#include <string>
#include <iostream>
#include <algorithm>

using namespace std;

int main()
{
    setlocale(LC_ALL, "Russian");

    secondTask();
}
```

Выше приведённый участок кода содержит директивы подключения необходимых в работе программы стандартных библиотек, объявление рабочего пространства имён, установку русской локализации обработки символов, основную функцию игры «Отгадай случайное число».

В функции `secondTask()` можно задать любой диапазон угадывания и любое количество попыток.

```
void secondTask()
{
    const unsigned int ATTEMPTS = 7; // количество попыток игрока
    const int SECRET_MIN = 1, SECRET_MAX = 100; // диапазон угадывания

    ...
}
```

Ниже представлен цикл, задающий основной интерфейс работы игры. До тех пор, пока пользователь решает продолжить игру после окончания очередной игровой сессии, игровая сессия перезапускается сначала.

```
...

do
{
    const int SECRET_NUMBER = getRandomInt(SECRET_MIN, SECRET_MAX);
    // секретное число, которое нужно угадать

    ...
}
```

Функция `getRandomInt()` генерирует число в требуемом диапазоне по алгоритму «Вихрь Мерсенна» версии «MT19937».

```
...

cout << "Начинаем игру \"Угадай число от "
      << SECRET_MIN << " до " << SECRET_MAX
      << " включительно\". У вас есть " << ATTEMPTS
      << " попыток, чтобы сделать это.\n";

bool player_won = playGame(ATTEMPTS, SECRET_NUMBER);
```

```

        if (player_won)
            cout << "Вы угадали загаданное число " << SECRET_NUMBER
                << ". Поздравляем!\n";
        else
            cout << "К сожалению, вы проиграли. Секретное число: "
                << SECRET_NUMBER << ".\n";

    } while (playAgain());

    cout << "Спасибо за игру!\n";
}

```

Игровые механики реализованы в функции playGame(), возвращающей логическое выражение. Если игрок победил, функция вернёт true, а иначе — false. После каждой попытки угадать число, игра подсказывает игроку, является ли введённое число искомым, большим или меньшим по отношению к искомому числу.

```

bool playGame(const unsigned int &ATTEMPTS, const int &SECRET_NUMBER)
{
    for (size_t counter = 1; counter <= ATTEMPTS; ++counter)
    {
        cout << "Ваша попытка №" << counter << ": ";

        int ATTEMPT; cin >> ATTEMPT;

        if (ATTEMPT > SECRET_NUMBER)
            cout << "Введённое число больше загаданного.\n";
        else if (ATTEMPT < SECRET_NUMBER)
            cout << "Введённое число меньше загаданного.\n";
        else
            return true; // игрок угадал число
    }

    return false; // игрок проиграл
}

```

Функция playAgain() реализовывает простой интерфейс обратной связи с игроком. Если игрок вводит символ «Y», функция возвращает true, если игрок вводит символ «N», функция возвращает false. От этого значения зависит работа while цикла в функции secondTask().

```

bool playAgain()
{
    // Спрашиваем у игрока, хочет ли он сыграть ещё раз

    char player_answer;

    do
    {
        cout << "Хотите сыграть ещё раз? (Y/N)\n";
        cin >> player_answer;

    } while (player_answer != 'Y' && player_answer != 'N');

    return (player_answer == 'Y');
}

```

Все описанные действия повторяются до тех пор, пока игрок не прекратит игру после очередной сессии, введя символ «N».

Описание переменных

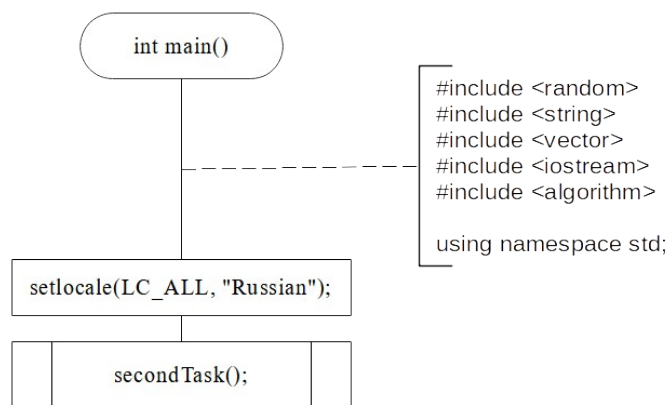
const unsigned int ATTEMPTS = 7; — количество попыток угадать число

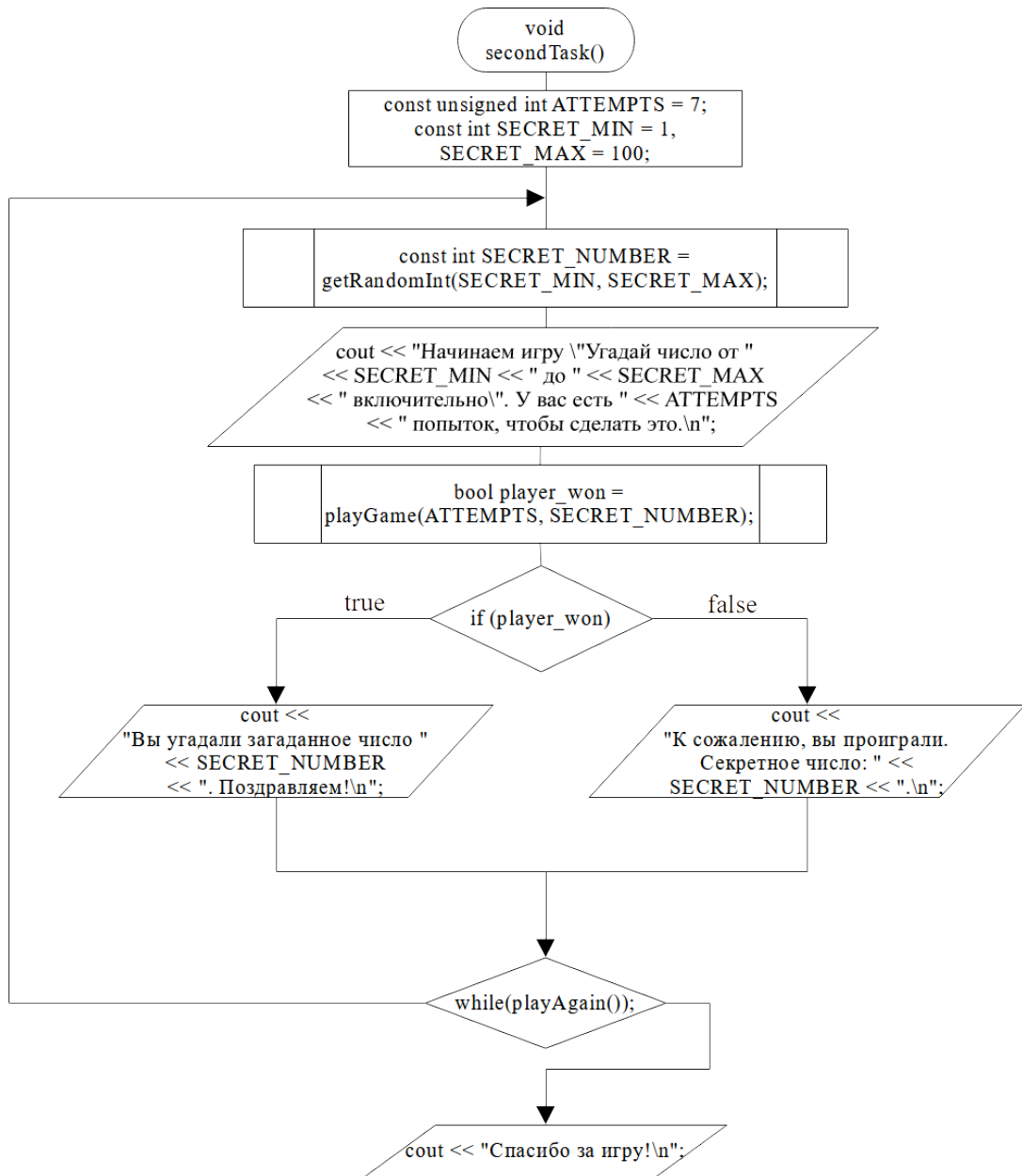
const int SECRET_MIN = 1,
SECRET_MAX = 100; — диапазон угадывания

const int SECRET_NUMBER =
getRandomInt(SECRET_MIN, SECRET_MAX); — число, которое нужно угадать

bool player_won =
playGame(ATTEMPTS, SECRET_NUMBER); — статус исхода игровой сессии

Блок-схема

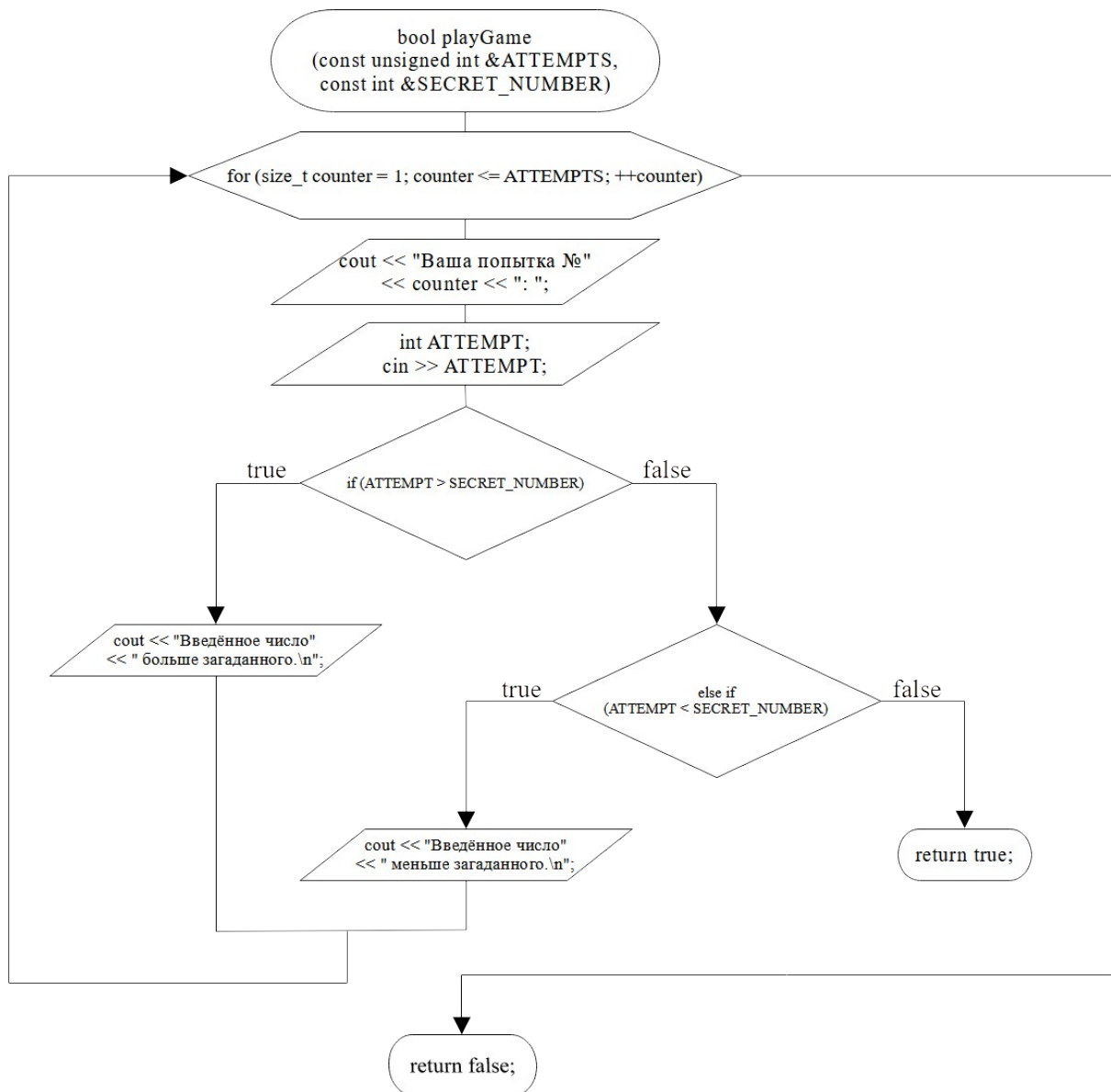


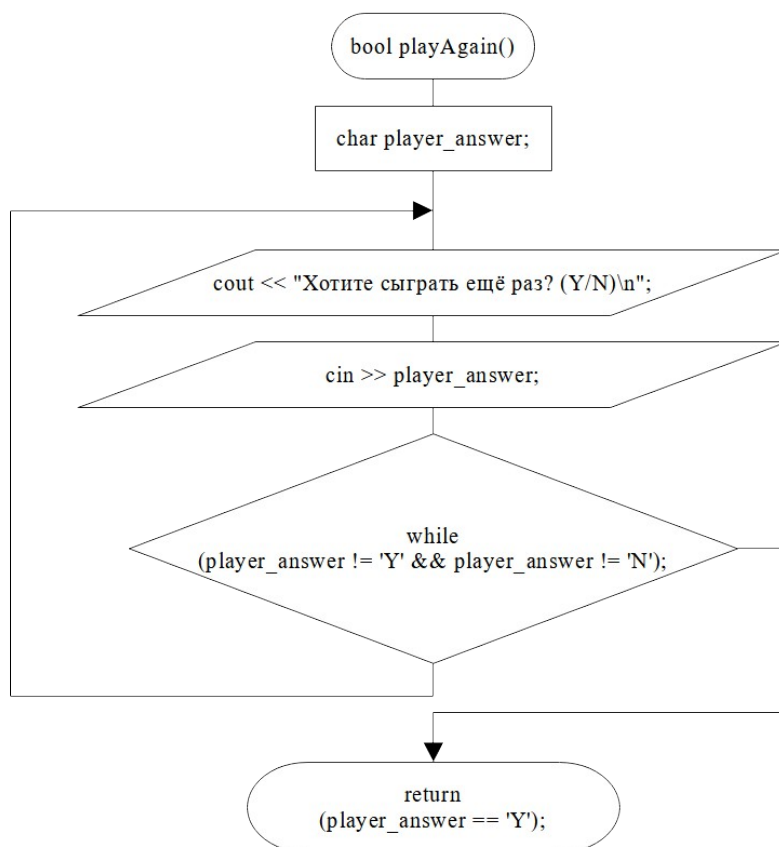


int getRandomInt
(const int &MIN, const int &MAX)

```
random_device random_device;  
mt19937 mersenne(random_device());  
uniform_int_distribution<> distribution(MIN, MAX);
```

return
distribution(mersenne);





Исходный код

```
/*
 * This code is licensed under the Creative Commons
 * Attribution - NonCommercial - NoDerivatives 4.0 International License.
 * To view a copy of this license, visit http://creativecommons.org/licenses/by-nc-nd/4.0/
 * or send a letter to Creative Commons, PO Box 1866, Mountain View, CA 94042, USA.
 *
 * https://bezukh.wixsite.com/blog
 * https://github.com/BezukhVladimir
 *
 * © Developed by Bezukh Vladimir. All right reserved.
 */

/*
 * Developed by Bezukh Vladimir
 * October 2021
 * ИБТ-21-26
 *
 * Реализовать игру «Угадай число».
 */

#include <random>
#include <string>
#include <iostream>
#include <algorithm>

using namespace std;

int getRandomInt(const int &MIN, const int &MAX)
{
    random_device random_device; // источник энтропии
    mt19937 mersenne(random_device()); // ГПСЧ «Вихрь Мерсенна»

    uniform_int_distribution<> distribution(MIN, MAX);

    return distribution(mersenne);
}

bool playGame(const unsigned int &ATTEMPTS, const int &SECRET_NUMBER)
{
    for (size_t counter = 1; counter <= ATTEMPTS; ++counter)
    {
        cout << "Ваша попытка №" << counter << ": ";

        int ATTEMPT; cin >> ATTEMPT;

        if (ATTEMPT > SECRET_NUMBER)
            cout << "Введённое число больше загаданного.\n";
        else if (ATTEMPT < SECRET_NUMBER)
            cout << "Введённое число меньше загаданного.\n";
        else
            return true; // игрок угадал число
    }

    return false; // игрок проиграл
}
```

```

bool playAgain()
{
    // Спрашиваем у игрока, хочет ли он сыграть ещё раз

    char player_answer;

    do
    {
        cout << "Хотите сыграть ещё раз? (Y/N)\n";
        cin >> player_answer;

    } while (player_answer != 'Y' && player_answer != 'N');

    return (player_answer == 'Y');
}

void secondTask()
{
    const unsigned int ATTEMPTS = 7; // количество попыток игрока
    const int SECRET_MIN = 1, SECRET_MAX = 100; // диапазон угадывания

    do
    {
        const int SECRET_NUMBER = getRandomInt(SECRET_MIN, SECRET_MAX);
        // секретное число, которое нужно угадать

        cout << "Начинаем игру \nУгадай число от "
              << SECRET_MIN << " до " << SECRET_MAX
              << " включительно\n". У вас есть " << ATTEMPTS
              << " попыток, чтобы сделать это.\n";

        bool player_won = playGame(ATTEMPTS, SECRET_NUMBER);

        if (player_won)
            cout << "Вы угадали загаданное число " << SECRET_NUMBER
                 << ". Поздравляем!\n";
        else
            cout << "К сожалению, вы проиграли. Секретное число: "
                 << SECRET_NUMBER << ".\n";

    } while (playAgain());

    cout << "Спасибо за игру!\n";
}

int main()
{
    setlocale(LC_ALL, "Russian");

    secondTask();
}

```

Скриншоты консольного интерфейса программы

```
Консоль отладки Microsoft Visual Studio
Начинаем игру "Угадай число от 1 до 100 включительно".
У вас есть 7 попыток, чтобы сделать это.
Ваша попытка №1: 50
Введённое число больше загаданного.
Ваша попытка №2: 25
Введённое число меньше загаданного.
Ваша попытка №3: 37
Введённое число больше загаданного.
Ваша попытка №4: 31
Введённое число больше загаданного.
Ваша попытка №5: 28
Введённое число больше загаданного.
Ваша попытка №6: 26
Введённое число меньше загаданного.
Ваша попытка №7: 27
Вы угадали загаданное число 27. Поздравляем!
Хотите сыграть ещё раз? (Y/N)
N
Спасибо за игру!
```

```
Консоль отладки Microsoft Visual Studio
Начинаем игру "Угадай число от -500 до 500 включительно".
У вас есть 10 попыток, чтобы сделать это.
Ваша попытка №1: 0
Введённое число больше загаданного.
Ваша попытка №2: -250
Введённое число больше загаданного.
Ваша попытка №3: -375
Введённое число больше загаданного.
Ваша попытка №4: -437
Введённое число меньше загаданного.
Ваша попытка №5: -406
Введённое число больше загаданного.
Ваша попытка №6: -421
Введённое число больше загаданного.
Ваша попытка №7: -429
Введённое число больше загаданного.
Ваша попытка №8: -433
Введённое число меньше загаданного.
Ваша попытка №9: -431
Введённое число меньше загаданного.
Ваша попытка №10: -430
Вы угадали загаданное число -430. Поздравляем!
Хотите сыграть ещё раз? (Y/N)
N
Спасибо за игру!
```

Анализ результатов

Все ранее заявленные и описанные особенности игрового процесса воспроизвелись корректно.