Федеральное государственное бюджетное образовательное учреждение высшего образования

**«ПЕРМСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

**ОТЧЕТ**

**по лабораторной работе**

Дисциплина «Основы алгоритмизации и программирования»

Семестр 1

Выполнил: студент группы РИС-21-1Б

Рыжков Никита Сергеевич

Проверил: доцент кафедры ИТАС

Полякова Ольга Андреевна

г. Пермь – 2021

**ПОСТАНОВКА ЗАДАЧИ №1**

Требуется создать программу на языке программирования C++, которая выводит все комбинации трехзначного числа, введенного пользователем, а также выводит самое большое число из списка получившихся чисел. Число, введенное пользователем, не должно содержать повторяющихся чисел.

**АНАЛИЗ РЕШЕНИЯ**

Для начала необходимо разделить задачу на блоки, поэтапное выполнение которых приведет к решению задачи. Итак, нам необходимо:

1. Получить число от пользователя;
2. Проверить, является ли введенное число трехзначным;
3. Выделить разряды числа;
4. Проверить, есть ли в числе повторяющиеся цифры;
5. Составить комбинации числа и вывести на экран;
6. Найти самое большое число и вывести на экран.

**Получение числа от пользователя.** Для того, чтобы получить необходимое нам трехзначное число, мы должны объяснить пользователю, какие введенные данные будут удовлетворять программу. Мы выводим сообщение *"Введите трехзначное число, в котором не будет повторяющихся чисел: ".* Далее ожидаем ввод данных от пользователя. Ввод и вывод на языке C++ реализуются с помощью команд *std::cin* и *std::cout* соответственно.

**Проверить, является ли введенное число трехзначным.** Для проверки числа мы сравниваем введенное число с 100 и с 999. Если оно меньше первого или больше второго, то пользователю необходимо ввести новое число. Данную проверку реализуем с помощью цикла с предусловием *while* и логического оператора || (или).

**Выделить разряды числа.** Данный шаг выполняется в четыре действия. Мы находим остаток от деления исходного числа на 10. Далее делим исходное число на 10, назовем его промежуточным. Потом мы находим остаток от деления промежуточного числа на 10. И в конце мы делим промежуточное число на 10. Таким образом, числа полученные в ходе в 1, 3, 4 действиях данного шага, являются разрядами числа. На языке данный шаг реализуется с помощью бинарных операторов % (остаток от деления) и / (целочисленное деление).

**Проверить, есть ли в числе повторяющиеся цифры.** Проверка выполняется поочередным сравнением разрядов числа. Если будет хотя бы одно совпадение, то пользователю будет необходимо ввести новое число. Реализация выполняется с помощью цикла *while* и двух операторов: == (равенство) и логического оператора || (или).

**Составить комбинации числа и вывести на экран.** Соберем числа из полученных ранее разрядов: abc, acb, bac, bca, cab, cba. Но необходимо еще проверить два последних разряда числа на равенство нулю и избавиться от чисел, где разряд, равный нулю стоит в начале, чтобы на выходе не получить такие значения: 097. Реализуем с помощью условного оператора *if-else*, оператора равенства (==) и оператора вывода *std::cout.*

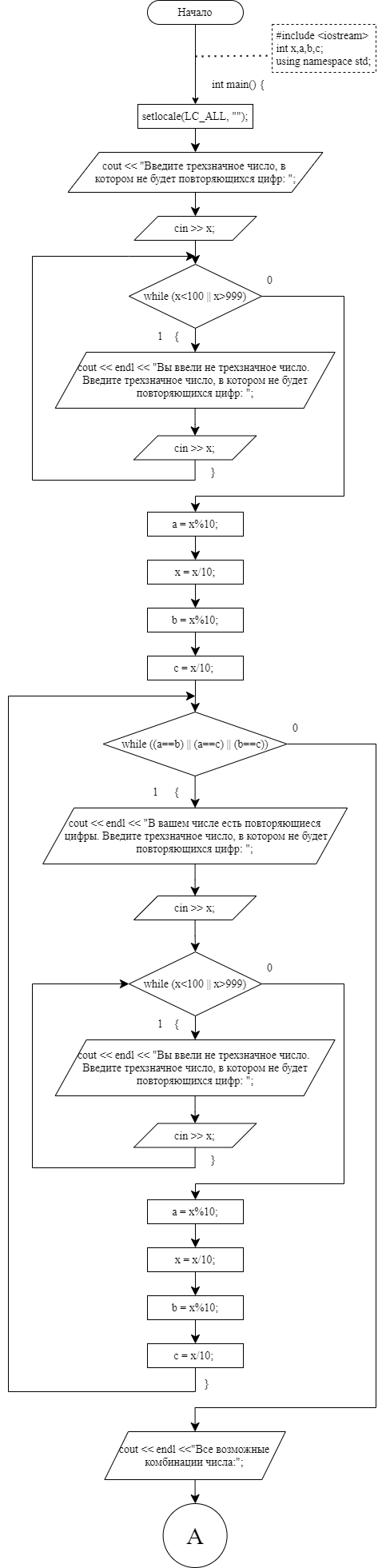
**Найти самое большое число и вывести на экран.** Для выполнения данного шага поочередно сравним разряды и выведем в порядке уменьшения. Реализация происходит за счет использования условного оператора if-else, логического оператора и (&&), оператора сравнения (>) и оператора вывода *std::cout*.

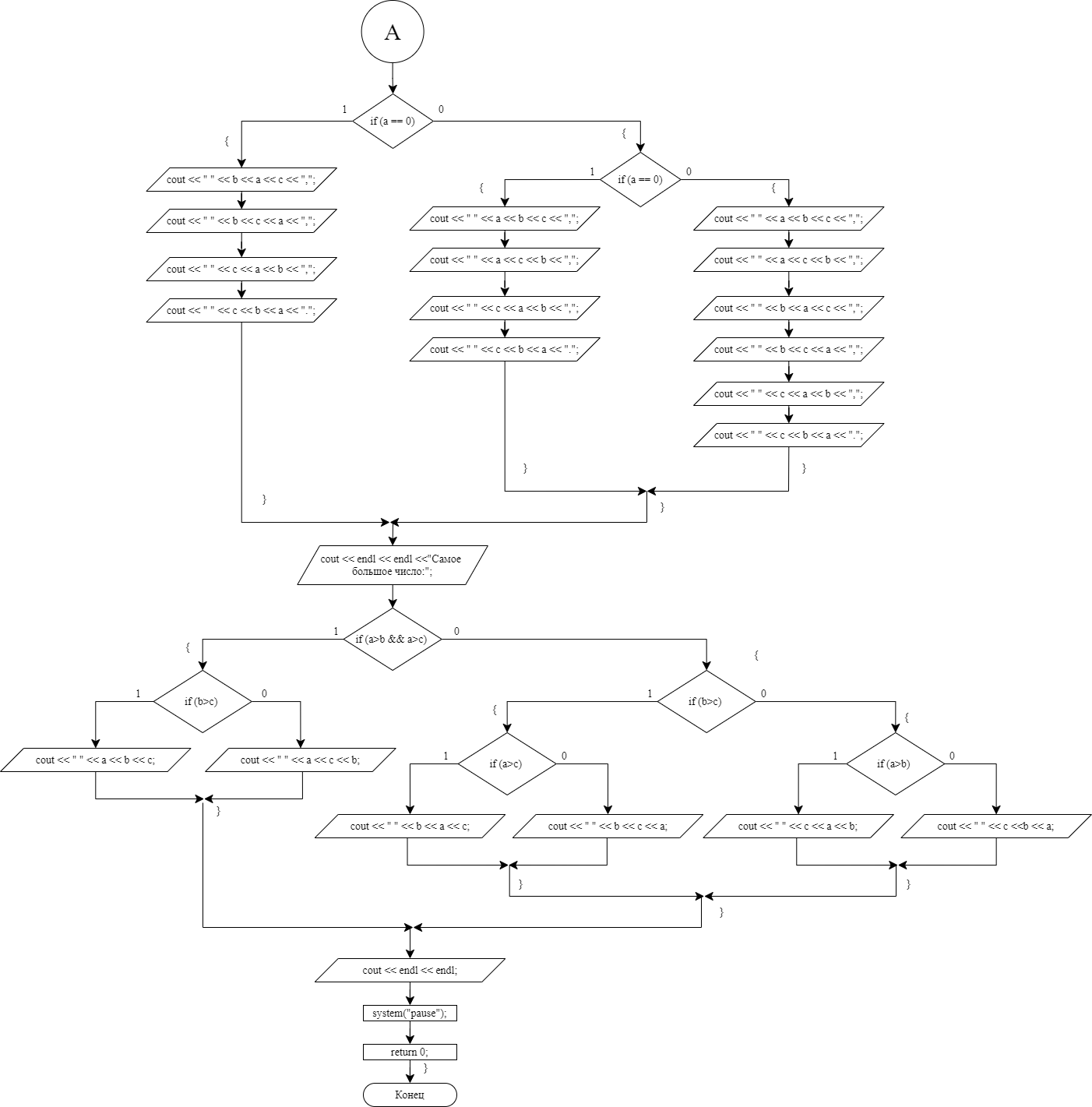
**ОПИСАНИЕ ПЕРЕМЕННЫХ**

*int* **x** – переменная целого типа данных. Используется для хранения числа, введенного пользователем.

*int* **a, b, c** – переменные целого типа данных. Используются для хранения разрядов числа, введенного пользователем.

**БЛОК-СХЕМА РЕШЕНИЯ**





**ЛИСТИНГ ПРОГРАММЫ**

#include <iostream>

int x,a,b,c;

using namespace std;

int main() {

setlocale(LC\_ALL, "");

cout << "Введите трехзначное число, в котором не будет повторяющихся цифр: ";

cin >> x;

while (x<100 || x>999) {

cout << endl <<"Вы ввели не трехзначное число. Введите трехзначное число, в котором не будет повторяющихся цифр: ";

cin >> x;

}

a = x%10;

x = x/10;

b = x%10;

c = x/10;

while ((a==b) || (a==c) || (b==c)) {

cout << endl <<"В вашем числе есть повторяющиеся цифры. Введите трехзначное число, в котором не будет повторяющихся цифр: ";

cin >> x;

while (x<100 || x>999) {

cout << endl <<"Вы ввели не трехзначное число. Введите трехзначное число, в котором не будет повторяющихся цифр: ";

cin >> x;}

a = x%10;

x = x/10;

b = x%10;

c = x/10;}

cout << endl <<"Все возможные комбинации числа:";

if (a == 0) {

cout << " " << b << a << c << ",";

cout << " " << b << c << a << ",";

cout << " " << c << a << b << ",";

cout << " " << c << b << a << ".";

}

else if (b == 0) {

cout << " " << a << b << c << ",";

cout << " " << a << c << b << ",";

cout << " " << c << a << b << ",";

cout << " " << c << b << a << ".";

}

else {

cout << " " << a << b << c << ",";

cout << " " << a << c << b << ",";

cout << " " << b << a << c << ",";

cout << " " << b << c << a << ",";

cout << " " << c << a << b << ",";

cout << " " << c << b << a << ".";

}

cout << endl << endl <<"Самое большое число:";

if (a>b && a>c) {

if (b>c) cout << " " << a << b << c;

else cout << " " << a << c << b;

}

else {

if (b>c) {

if (a>c) cout << " " << b << a << c;

else cout << " " << b << c << a;

}

else {

if (a>b) cout << " " << c << a << b;

else cout << " " << c << b << a;

}}

cout << endl << endl;

system("pause");

return 0;}

**РЕЗУЛЬТАТЫ**

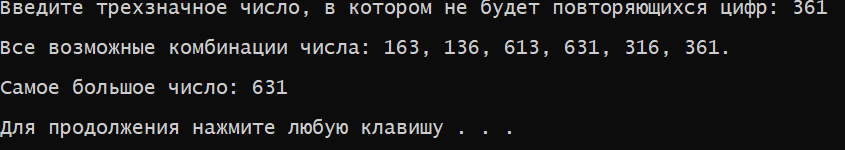


Рис. 1 – Результат программы после ввода корректного числа

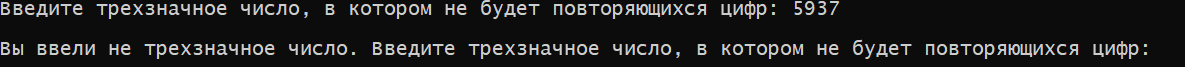


Рис. 2 – Результат программы после ввода четырехзначного числа

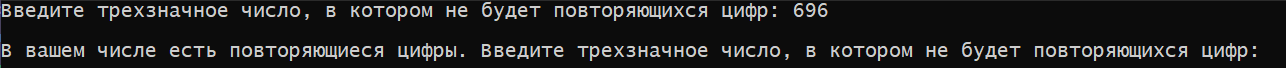


Рис. 3 – Результат программы после ввода числа с повторяющимися цифрами

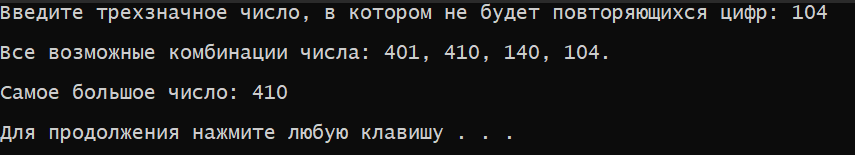


Рис. 4 – Результат программы после ввода числа, содержащего ноль.

**АНАЛИЗ РЕЗУЛЬТАТОВ**

Как показано на рисунках 1 и 4, программа успешно справляется с поставленными задачами при вводе трехзначного числа без повторяющихся цифр. Но если в первом случае получилось 6 комбинаций, то в последнем только 4, так как когда 0 стоит в начале числа, он теряет смысл (014 = 14). Пример ввода четырехзначного числа проиллюстрирован на рисунке 2. Программа требует от пользователя другое число, подходящее условию. На рисунке 3 можно наблюдать ввод числа с повторяющимися цифрами. Число также не подходит условию, поэтому выводится сообщение об ошибке и возможность повторного ввода числа.

**ПОСТАНОВКА ЗАДАЧИ №2**

Требуется создать программу на языке программирования C++, которая случайно генерирует число от 1 до 100, в ответ выводится сообщение о том, больше ли загаданное число, чем введенное или меньше. На отгадывание дается 7 попыток, в случае неудачи выводится соответствующее сообщение.

**АНАЛИЗ РЕШЕНИЯ**

Для начала необходимо разделить задачу на блоки, поэтапное выполнение которых приведет к решению задачи. Итак, нам необходимо:

1. Получить случайное число;
2. Получить число от пользователя;
3. Проверить оставшиеся попытки;
4. Проверить, отгадал ли пользователь число или нет;
5. Вывод промежуточных и конечных итогов;

**Получить случайное число.** Нам необходимо придумать число в пределах от 1 до 100 включительно. На C++ данная задача решается с помощью объединения функций *srand(), rand(), time(),* которое позволяет создать случайное число, которое не будет повторятся при следующих запусках программы

**Получить число от пользователя.** Для того, чтобы получить необходимое нам трехзначное число, мы должны объяснить пользователю, какие введенные данные будут удовлетворять программу. Мы выводим сообщение *" Я загадал число от 1 до 100. Попробуй угадать! Введите свой вариант: ".* Далее ожидаем ввод данных от пользователя. Ввод и вывод на языке C++ реализуются с помощью команд *std::cin* и *std::cout* соответственно.

**Проверить оставшиеся попытки.** Нам необходимо создать счетчик попыток, который будет прибавлять единицу после каждой попытки. А также нужно проверять, больше ли счетчик семи. Программно реализуем с помощью цикла *while* и постфиксного инкремента *(count++).*

**Проверить, отгадал ли пользователь число или нет.** Данный шаг выполняется с помощью сравнения числа, введенного пользователем и числа, сгенерированного программой. Реализуется это действие с помощью условного оператора *if-else* и операторов сравнения больше-меньше (<>).

**Вывод промежуточных и конечных итогов.** По мере отгадывания пользователем числа, нам необходимо выводит сообщения, которые расскажут ему больше или меньше введенное им число, а также угадал ли он число. Вывод на языке C++ реализуются с помощью команд *std::cout*.

**ОПИСАНИЕ ПЕРЕМЕННЫХ**

*int* **x** – переменная целого типа данных. Используется для хранения числа, введенного пользователем.

*int* **number** – переменная целого типа данных. Используются для хранения числа, сгенерированного программой.

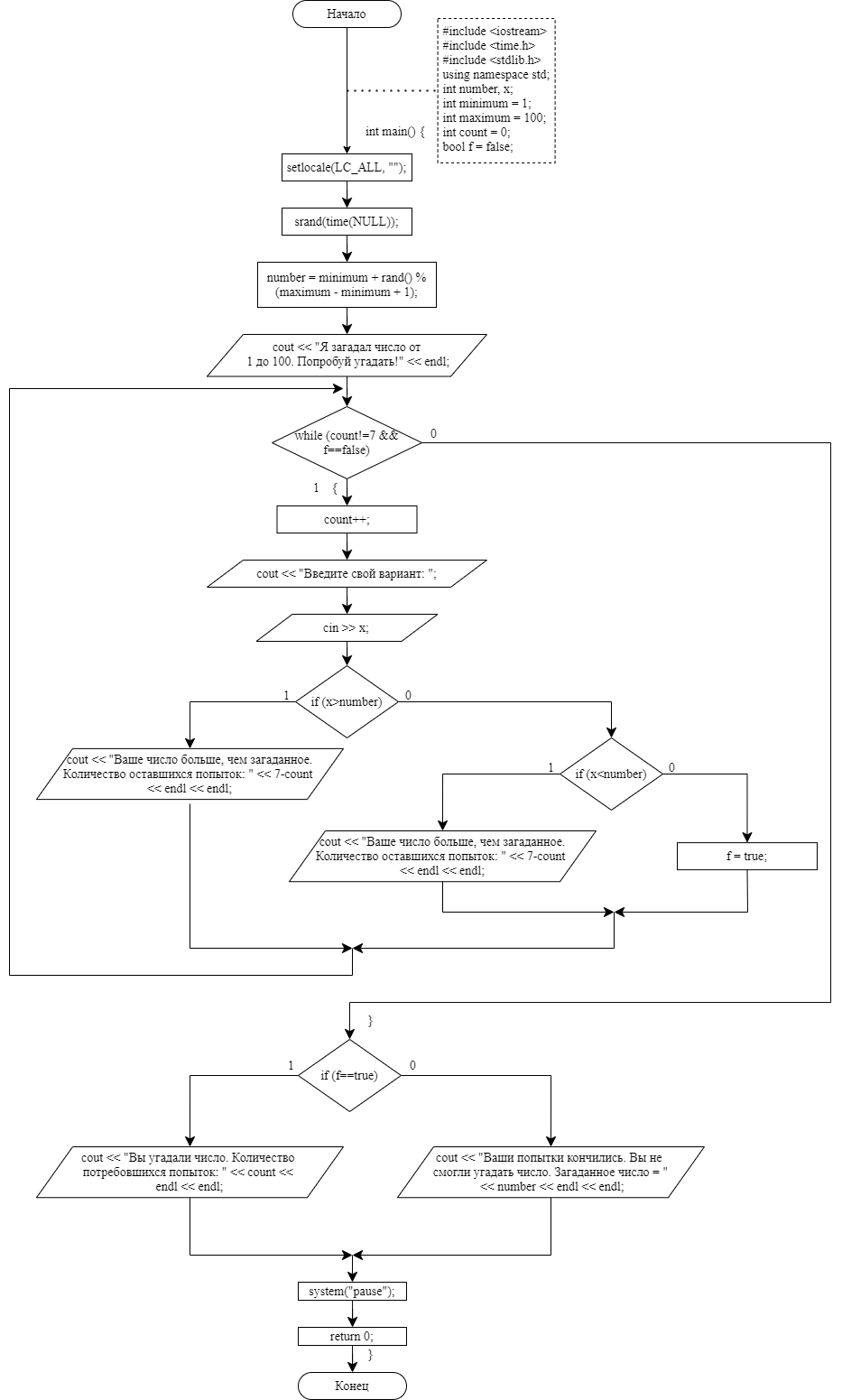
int **minimum = 1** – переменная целого типа данных. Используются для хранения числа, являющегося минимумом для генерации числа.

*int* **maximum = 100** – переменная целого типа данных. Используются для хранения числа, являющегося максимумом для генерации числа.

*int* **count = 0** – переменная целого типа данных. Используются для хранения числа, равного текущему количеству попыток.

bool **f = false** – переменная логического типа данных. Используются для определения, отгадал ли пользователь число или нет.

**БЛОК-СХЕМА РЕШЕНИЯ**



**ЛИСТИНГ ПРОГРАММЫ**

#include <iostream>

#include <time.h>

#include <stdlib.h>

using namespace std;

int number, x;

int minimum = 1;

int maximum = 100;

int count = 0;

bool f = false;

int main() {

setlocale(LC\_ALL, "");

srand(time(NULL));

number = minimum + rand() % (maximum - minimum + 1);

cout << "Я загадал число от 1 до 100. Попробуй угадать!" << endl;

while (count!=7 && f==false) {

count++;

cout << "Введите свой вариант: ";

cin >> x;

if (x>number) cout << "Ваше число больше, чем загаданное. Количество оставшихся попыток: " << 7-count << endl << endl;

else if (x<number) cout << "Ваше число меньше, чем загаданное. Количество оставшихся попыток: " << 7-count << endl << endl;

else f = true;

}

if (f==true) cout << "Вы угадали число. Количество потребовшихся попыток: " << count << endl << endl;

else cout << "Ваши попытки кончились. Вы не смогли угадать число. Загаданное число = " << number << endl << endl;

system("pause");

return 0;

}

**РЕЗУЛЬТАТЫ**

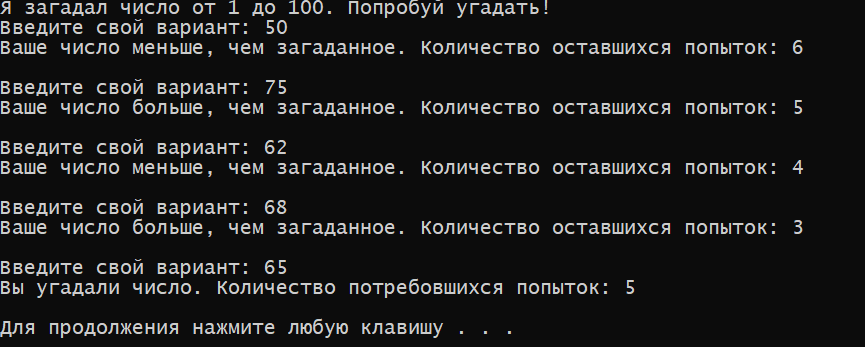


Рис. 5 – Результат программы, когда пользователь угадал число

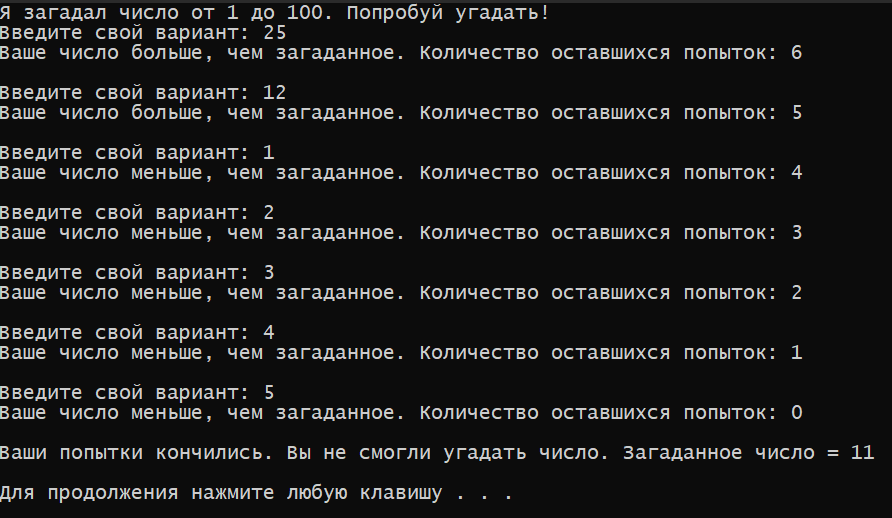


Рис. 6 – Результат программы, когда пользователь не угадал число

**АНАЛИЗ РЕЗУЛЬТАТОВ**

Как мы видим на рисунках 5 и 6, программа создает разные числа при повторном запуске. В обоих случаях она выводит подсказку и количество оставшихся попыток после ввода пользователем числа. При исчерпании попыток программа выводит загаданное число.