

ФИО: Ребдев Павел Александрович  
Группа: 5130904/30008  
Лабораторная работа: «Функции. Простейшее»

### Постановка задачи

Разработать детальные требования и тест план для следующей задачи:

**Цифра в числе** Функция, имеющая два параметра (целые числа N1 и N2) и возвращающая значение true, если десятичная цифра, заданная первым параметром, встречается в десятичной записи числа, заданного вторым параметром. Незначащие нули не учитываются

### Детальные требования

1. Число N1 должно быть задано корректно:
  - 1.1. N1 — цифра от 0 до 9. Если N1 не является цифрой от 0 до 9, то выводится сообщение об ошибке и программа завершается
2. Число N2 должно быть заданы корректно:
  - 2.1. N2 — число. Если N2 не является числом, то выводится сообщение об ошибке и программа завершается
3. В числе N2 встречается символ N1:
  - 3.1. Если в числе N2 встречается N1, то программа завершается с кодом возврата 0 и сообщением: «1»
  - 3.2. Если в числе N2 не встречается N1, то программа завершается с кодом возврата 0 и сообщением: «0»

### Тест-план

Проверка детальных требований с помощью тест-плана:

#	Описание	Результат
1.1	N1 — цифра от 0 до 9. Если N1 не является цифрой от 0 до 9, то выводится сообщение об ошибке и программа завершается	<b>Input:</b> 12 123 <b>Expected:</b> Input Error! First argument must be digital from 0 to 9
2.1	N2 — число. Если N2 не является числом, то выводится сообщение об ошибке и программа завершается	<b>Input:</b> 2 abc <b>Expected:</b> Input error! Second argument must be number
3.1	Если в числе N2 встречается N1, то программа завершается с кодом возврата 0 и сообщением: «1»	<b>Input:</b> 2-12 <b>Expected:</b> 1
3.1		<b>Input:</b> 4 1441 <b>Expected:</b> 1
3.2	Если в числе N2 не встречается N1, то программа завершается с кодом возврата 0 и сообщением: «0»	<b>Input:</b> 1 232 <b>Expected:</b> 0
		<b>Input:</b> 3 14152 <b>Expected:</b> 0

### Исходные тексты программы

Файлы с исходными текстами лабораторной работы (полагаем <ROOT> для папки в котором располагаются исходные тексты):

./<ROOT>/main.cpp

```
#include <iostream>
#include <stdexcept>
#include "digitInNumber.hpp"
int main ()
{
    unsigned int digital = 0;
    long long int number = 0;

    std::cin >> digital;
    if (!std::cin)
    {
        std::cerr << "Input Error! First argument must be digital from 0 to 9" << "\n";
        return 1;
    }

    std::cin >> number;
    if (!std::cin)
    {
        std::cerr << "Input error! Second argument must be number\n";
        return 1;
    }

    try
    {
        std::cout << isDigitalInNumber(digital, number) << '\n';
    }
    catch (const std::invalid_argument & e)
    {
        std::cerr << e.what() << '\n';
        return 1;
    }
    return 0;
}
```

**./<ROOT>/digitInNumber.cpp**

```
#include "digitInNumber.hpp"
#include <stdexcept>

bool isDigitalInNumber (unsigned int digital, long long int number)
{
    if (number < 0)
    {
        number *= -1;
    }
    if (digital > 9)
    {
        throw std::invalid_argument ("Digital can't be more then 9");
    }
    while (number != 0)
    {
        if ((number % 10) == digital)
        {
            return 1;
        }
        number /= 10;
    }
    return 0;
}
```

**./<ROOT>/digitInNumber.hpp**

```
#ifndef DIGITALINNUMBER_HPP
#define DIGITALINNUMBER_HPP
bool isDigitalInNumber (unsigned int digital, long long int number);
#endif
```