ФИО: Ребдев Павел Александрович
Группа: 5130904/30008
Лабораторная работа: «Строки»

Постановка задачи

Разработать детальные требования и тест план для следующей задачи:

[DGT-SND]. Реализовать программу формирующую строку из двух исходных, добавив в первую все символы, встречающиеся во второй и являющиеся десятичными цифрами

Детальные требования

1. Длина строки должна быть задана корректно:
   1.1. Если строка пуста, то программа завершается с кодом возврата 1 и соответствующим сообщением об ошибке в стандартном потоке ошибок
   1.2. Если строка имеет слишком большую длину, то программа завершается с кодом возврата 1 и соответствующим сообщением об ошибке в стандартном потоке ошибок
2. Длина строки задана корректно:
   2.1. Если длина строки задана корректно, на завершается с кодом возврата 0 и сообщением в стандартный поток вывода, состоящем из строки, которая является добавлением всех десятичных цифр из второй строки в первую

Тест-план

Проверка детальных требований с помощью тест-плана:

#	Описание	Результат
1.1	Если строка пуста, то программа завершается с кодом возврата 1 и соответствующим сообщением об ошибке в стандартном потоке ошибок	**Input:** **Expected:** `String is empty` **exit code:** `1`

1.2	Если строка имеет слишком большую длину, то программа завершается с кодом возврата 1 и соответствующим сообщением об ошибке в стандартном потоке ошибок	**Input:**
		`(строка из 100000000 элементов)`
		Expected:
		`String is very big`
		exit code: `1`

2.1	Если длина строки задана корректно, на завершается с кодом возврата 0 и сообщением в стандартный поток вывода, состоящем из строки, которая является добавлением всех десятичных цифр из второй строки в первую	**Input:**
		`Some string`
		Second string:
		`1235  892`
		Expected:
		`Some string1235892`
		exit code: `0`

		Input:
		`String number one`
		Second string:
		`some random text`
		Expected:
		`String number one`
		exit code: `0`

		Input:
		`String 1235`
		Second string:
		`OK 3 – 12 + 149`
		Expected:
		`String 1235312149`
		exit code: `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 <cstddef>
#include <cstring>
#include "stringInput.hpp"
#include "stringConvert.hpp"


int main()
{
  char * firstStr = nullptr;
  size_t firstSize = 0;
  try
  {
    firstStr = rebdev::acceptStr(std::cin, firstSize);
  }
  catch (const std::exception & e)
  {
    delete[] firstStr;
    std::cerr << e.what() << '\n';
    return 1;
  }
  if (firstStr == nullptr)
  {
    delete[] firstStr;
    std::cerr << "String is empty\n";
    return 1;
  }

  char const * const secondStr = "1 2ok 3 5z 3pv21";

  size_t numOfDig = 0;
  for (size_t i = 0; secondStr[i] != '\0'; ++i)
  {
    if (std::isdigit(secondStr[i]))
    {
      numOfDig += 1;
    }
  }

  char * finishStr = nullptr;
  try
  {
    finishStr = new char[firstSize + numOfDig + 1];
    for (size_t i = 0; i < (firstSize + numOfDig); ++i)
    {
      finishStr[i] = '0';
    }
    finishStr[firstSize + numOfDig] = '\0';
  }
  catch (const std::exception & e)
  {
    return 2;
  }
```

```
  rebdev::addNumbers(firstStr, secondStr, finishStr);

  std::cout << finishStr << '\n';
  delete[] finishStr;
  delete[] firstStr;

  return 0;
}
```

./<ROOT>/stringConvert.hpp

```
#ifndef STRINGCONVERT_HPP
#define STRINGCONVERT_HPP
namespace rebdev
{
   void addNumbers(const char * firstString, const char * secondString,
char * endString);
}
#endif
```

./<ROOT>/stringConvert.cpp

```
#include "stringConvert.hpp"
#include <cstddef>
#include <cctype>

void rebdev::addNumbers(const char * firstString, const char *
secondString, char * endString)
{
  size_t i = 0;
  while (firstString[i] != '\0')
  {
    endString[i] = firstString[i];
    i += 1;
  }
  size_t j = 0;
  while (secondString[j] != '\0')
  {
    if (std::isdigit(secondString[j]))
    {
      endString[i] = secondString[j];
      i += 1;
    }
    j += 1;
  }
}
```

./<ROOT>/stringInput.hpp

```
#ifndef STRINGCONVERT_HPP
#define STRINGCONVERT_HPP
namespace rebdev
{
   void addNumbers(const char * firstString, const char * secondString,
char * endString);
}
#endif
```

./<ROOT>/stringInput.cpp

```
#include "stringInput.hpp"

char * rebdev::acceptStr(std::istream & input, size_t & sizeOfStr)
{
  char sym = 0;
  sizeOfStr = 0;
  size_t sizeOfBuffer = 0;

  char * str = nullptr;
  char * str2 = nullptr;

  input >> std::noskipws;

  while (input >> sym)
  {
    if (!input)
    {
      delete[] str;
      input >> std::skipws;
      throw std::logic_error("Bad read!");
    }

    if (sizeOfStr == sizeOfBuffer)
    {
      sizeOfBuffer += 10;
      try
      {
        str2 = new char[sizeOfBuffer];
      }
      catch (const std::exception & e)
      {
        input >> std::skipws;
        delete[] str;
        delete[] str2;
        throw;
      }

      for (size_t i = 0; i < sizeOfStr; ++i)
      {
        str2[i] = str[i];
      }
       for (size_t i = sizeOfStr; i < (sizeOfBuffer * (str != nullptr));
++i)
      {
        str2[i] = '\0';
      }

      delete[] str;
      str = str2;
      str2 = nullptr;
    }
```

```
    str[sizeOfStr] = sym;
    sizeOfStr += 1;
  }

  input >> std::skipws;
  delete[] str2;
  return str;
}
```