**САНКТ-ПЕТЕРБУРГСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ ПЕТРА ВЕЛИКОГО**

**ИНСТИТУТ КОМПЬЮТЕРНЫХ НАУК И КИБЕРБЕЗОПАСНОСТИ**

**ВЫСШАЯ ШКОЛА ПРОГРАММНОЙ ИНЖЕНЕРИИ**

**Индивидуальное задание №2**

**Вариант 2, 11**

**«Строки в стиле C»**

**по предмету «Алгоритмизация и программирование»**

**Выполнил: студент гр. 5130904/30002 Севостьянова А.В.**

**Руководитель: Череповский Д.К.**

**Санкт-Петербург**

**2023 г.**

Оглавление

[1. Общая постановка задачи 3](#_Toc149955310)

[2. Детальные требования и тест план 3](#_Toc149955311)

[Длина обрабатываемой строки(length) должна быть задана корректно (натуральное положительное число) 3](#_Toc149955312)

[Выбор пользователя (choice) должен быть задан корректно(целое число равное 1 или 2) 3](#_Toc149955313)

[Таблица с детальными требованиями и тест планом 4](#_Toc149955314)

[3. Программа 5](#_Toc149955315)

[Main.cpp 5](#_Toc149955316)

[Str\_Functions.h 7](#_Toc149955317)

[Str\_Functions.cpp 7](#_Toc149955318)

[4. Результат работы программы 8](#_Toc149955319)

[Приложение A 9](#_Toc149955320)

[Приложение B 10](#_Toc149955321)

[Вывод 12](#_Toc149955322)

1. **Общая постановка задачи**

Написать программу, считывающую из исходного файла длину строки и исходную строку и применяющую к данной строке одну из следующих функций, реализованных в стиле C:

1. *функцию, удаляющую все десятичные цифры из исходной строки.*

*2. Функцию, заменяющую все прописные латинские буквы в исходной строке на строчные.*

Результат работы функции должен быть выведен в выходной файл.

**Входные данные программы:** файл, содержащий длину строки и строку, номер выбранной функции.

**Выходные данные программы:** преобразованная строка

**Входные данные функции**: адрес исходной строки(destination), адрес полученной строки (source)

**Выходные данные функции:** указатель на строку результата

1. **Детальные требования и тест план**

## *Длина обрабатываемой строки(length) должна быть задана корректно (натуральное положительное число)*

1. Если length принимает нечисловое значение, то сообщение: «Длина строки задана некорректно», аварийное завершение программы
2. Если length принимает дробное значение, то сообщение «Длина строки задана некорректно», аварийное завершение программы
3. Если length принимает отрицательное значение или равна 0, то сообщение «Длина строки задана некорректно», аварийное завершение программы

## *Выбор пользователя (choice) должен быть задан корректно(целое число равное 1 или 2)*

1. Если choice принимает нечисловое значение, то сообщение: «Некорректный ввод», аварийное завершение программы
2. Если choice принимает дробное значение, то сообщение: «Некорректный ввод», аварийное завершение программы
3. Если choice принимает значение равное 1, то вызов функции deleteDigitsFromStr
4. Если choice принимает значение равное 1, то вызов функции convertToLowercase
5. Если choice принимает значение не равное 1 или 2, то сообщение: «Некорректный ввод», аварийное завершение программы.

## *Таблица с детальными требованиями и тест планом*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Требования | Детальные требования | | Данные | | Ожидаемый результат | |
| 1. ***Длина обрабатываемой строки должна быть задана корректно*** | | | | | | |
| 1. length - число | | Если length имеет нечисловое значение, то сообщение:  «Длина строки задана некорректно», аварийное завершение программы | | R | | Сообщение:  «Длина строки задана некорректно»  Аварийное завершение программы. |
| 2. length – целое число | | Если length имеет дробное значение, то сообщение: сообщение: «Длина строки задана некорректно», аварийное завершение программы | | 5,5 | | Сообщение:  «Длина строки задана некорректно»  Аварийное завершение программы. |
| 3. length > 0 | | Если length <= 0 сообщение: «Длина строки задана некорректно», аварийное завершение программы | | -1 | | Сообщение:  «Длина строки задана некорректно»  Аварийное завершение программы. |
| 4. length задана корректно | | Если length натуральное неотрицательное число | | 5 | | Продолжение работы программы. |
| 1. ***Выбор пользователя должен быть корректен*** | | | | | | |
| 1. choice - число | Если choice имеет нечисловое значение, то сообщение: «Некорректный ввод», аварийное завершение программы. | | R | | Сообщение: «Некорректный ввод»  Аварийное завершение программы. | |
| 2. choice – целое число | Если choice имеет дробное значение, то сообщение: «Некорректный ввод», аварийное завершение программы. | | 5,5 | | Сообщение: «Некорректный ввод»  Аварийное завершение программы. | |
| 3. choice == 1 | Если choice = 1, вызов функции deleteDigitsFromStr | | 1 | | Вызов функции deleteDigitsFromStr. | |
| 4. choice == 2 | Если choice = 2, вызов функции convertToLowercase | | 2 | | Вызов функции convertToLowercase | |
| 4. choice != 1  и choice != 2 | Если choice != 1  и choice != 2, то сообщение: «Некорректный ввод», аварийное завершение программы. | | 5 | | Сообщение: «Некорректный ввод»  Аварийное завершение программы. | |

# 3. Программа

## *Main.cpp*

#include<fstream>

#include"C\_str\_Functions.h"

using namespace std;

const string IN\_FILE = "in.txt";

const string OUT\_FILE = "out.txt";

const string ERR\_OF\_OPENING = "Ошибка открытия файла";

const string ERR\_OF\_TYPE = "Длина строки задана некорректно";

int main()

{

int result = EXIT\_SUCCESS;

setlocale(LC\_ALL, "ru");

ifstream in(IN\_FILE);

ofstream out(OUT\_FILE);

char \*strFileIN = nullptr; // исходная строка

char \*strFileOut = nullptr; // полученная строка

try

{

if (!in.is\_open() || !out.is\_open())

throw ios\_base::failure(ERR\_OF\_OPENING);

int length = 0;

in >> length;

if (in.fail() || in.peek() != '\n'|| length<=0)

throw invalid\_argument(ERR\_OF\_TYPE);

strFileIN = new char[length + 1] ();

in.ignore();

in.read(strFileIN, length);

cout << "Исходная строка\n (Если исходная строка была больше заданной длины, то она была обрезана)\n";

showStr(strFileIN);

cout << "Выберите функцию\n 1 - Удаление всех цифр\n 2 - Замена прописных букв строчными\nВыбор:\t";

string value = "";

cin >> value;

int choice = checkInput(value);

strFileOut = new char[length + 1] ();

switch (choice)

{

case 1:

deleteDigitsFromStr(strFileOut, strFileIN);

out << strFileOut << "\n";

cout << "Полученная строка:\t";

break;

case 2:

convertToLowercase(strFileOut, strFileIN);

out << strFileOut << "\n";

cout << "Полученная строка:\t";

break;

default:

cerr << ERROR << "\n";

result = EXIT\_FAILURE;

}

}

catch(ios\_base::failure &ex)

{

cerr << ex.what() << "\n";

result = EXIT\_FAILURE;

}

catch (invalid\_argument &ex)

{

cerr << ex.what() << "\n";

result = EXIT\_FAILURE;

}

catch (bad\_alloc& ex)

{

cerr << ex.what() << "\n";

result = EXIT\_FAILURE;

}

in.close();

out.close();

delete[]strFileIN;

delete[]strFileOut;

return result;

}

## *Str\_Functions.h*

#pragma once

#include<iostream>

#include<regex>

#include<string>

const std::string ERROR = "Некорректный ввод";

char \* deleteDigitsFromStr(char \*destination, const char \*source);

char\* convertToLowercase(char\* destination, const char\* source);

void showStr(const char\* str);

int checkInput( const std::string &value);

## *Str\_Functions.cpp*

#include "C\_Str\_Functions.h"

char\* deleteDigitsFromStr(char \*destination, const char \*source)

{

int i = 0;

int j = 0;

while (source[i] != '\0')

{

if (!isdigit(source[i]))

{

destination[j] = source[i];

j++;

}

i++;

}

destination[j] = '\0';

return destination;

}

char\* convertToLowercase(char\* destination, const char\* source)

{

int i = 0;

while (source[i] != '\0')

{

if (isalpha(source[i]) && isupper(source[i]))

destination[i] = tolower(source[i]);

else destination[i] = source[i];

i++;

}

destination [i] = '\0';

return destination;

}

void showStr(const char\* str)

{

int i = 0;

while (str[i] != '\0')

{

std::cout << str[i];

i++;

}

std::cout << "\n\n";

}

int checkInput(const std::string &value)

{

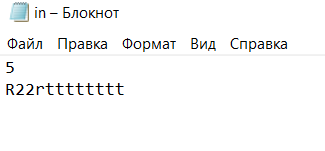
std::regex regular ("\\d+");

if (!std::regex\_match(value, regular)) throw std::invalid\_argument(ERROR);

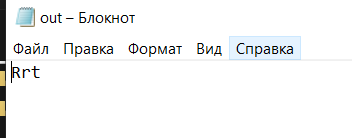
return stoi(value);

}

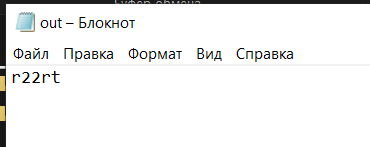
# 4. Результат работы программы



*Рис. 1 Исходный файл*



*Рис. 2 Полученный файл после выполнения функции, удаляющей десятичные цифры из строки*

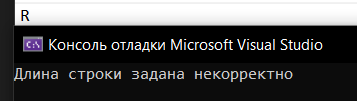


*Рис. 3 Полученный файл после выполнения функции, переводящей символы верхнего регистра в нижний*

# Приложение A

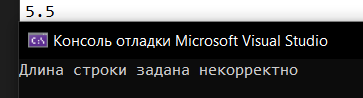
***Длина обрабатываемой строки(length) должна быть задана корректно (натуральное положительное число)***

1. length - число



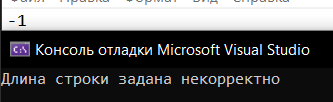
*Рис. 4 length принимает нечисловое значение*

1. Length – целое число



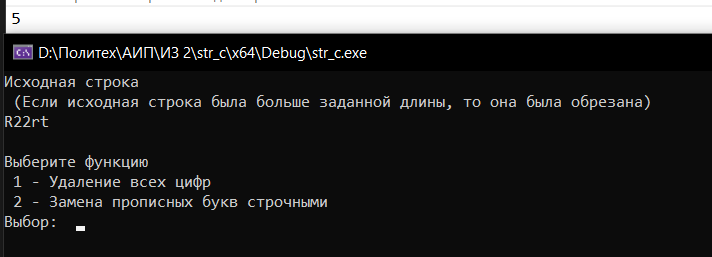
*Рис. 5 length принимает дробное значение*

1. Length – натуральное неотрицательное число



*Рис. 6 length принимает отрицательное значение или равна 0*

1. Length – задана корректно

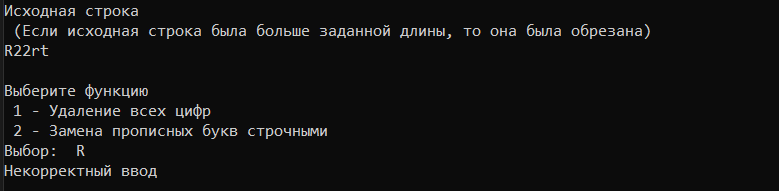


*Рис. 7 length задана корректно*

# Приложение B

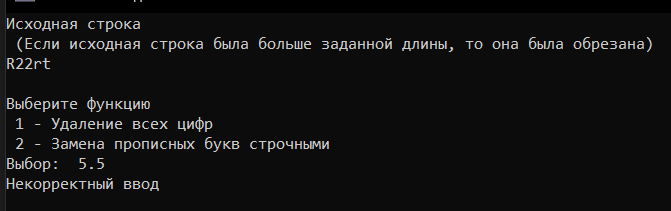
***Выбор пользователя (choice) должен быть задан корректно(целое число равное 1 или 2)***

1. choice - число



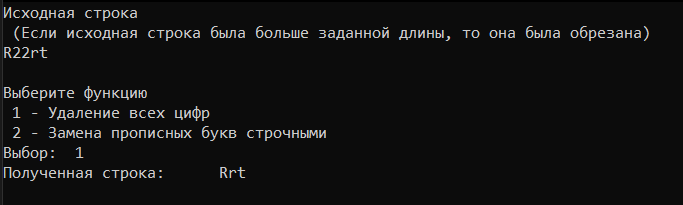
*Рис. 8 choice принимает нечисловое значение*

1. choice – целое число



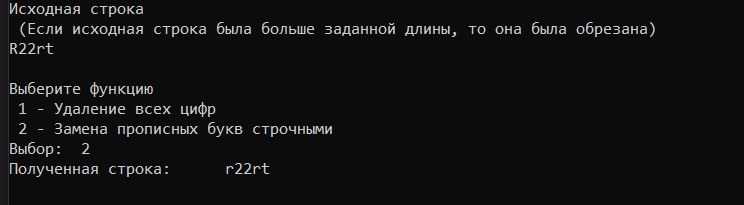
*Рис. 9 choice принимает дробное значение*

1. choice == 1



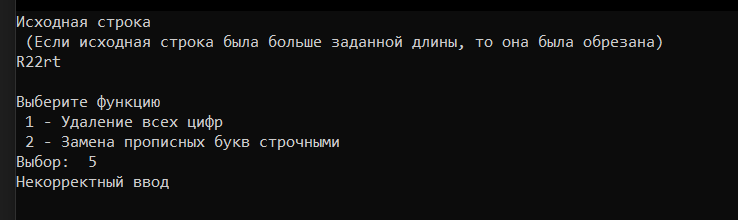
*Рис. 10 choice == 1*

1. choice == 2



*Рис. 11 choice == 2*

5. choice != 1 и choice != 2



*Рис. 12 choice != 1 и choice!=2*

# Вывод

В ходе работы над программой

1. Мы узнали о строках в стиле C и научились с ними работать
2. Выполняемый код был разделе на логические части - функции
3. Научились работать с файлами и обрабатывать возникающие при работе ошибки
4. Использовали регулярные выражения для проверки корректности ввода данных
5. Использовали оператор try … catch для обработки возникающих ошибок
6. Использовали исключения, что позволяет эффективнее обрабатывать возникающие в ходе выполнения программы ошибки