**Лабораторная работа № 7**

**Задание № 3**

1. **Общая постановка задачи**

Сформировать новую строку, удалив из исходной строки все буквы латинского алфавита.

1. **Таблица с детальными требованиями и тест план**

|  |  |  |  |
| --- | --- | --- | --- |
| **Требование** | **Детальные требования** | **Данные** | **Ожидаемый результат** |
| 1. **Данные должны быть введены корректно** | | | |
| 1.1  Длина первой строки – целое положительное число | Если введено не целое число, отрицательное число или ноль, сообщение: "Длина строки – положительное число!". | А  9.7 | Сообщение: «Длина строки – положительное число!» |
| 1.2 В директории проекта должен существовать файл со входными данными с корректным названием. | Если его нет, сообщение: "Файл не найден" | Inyd.txt | Сообщение: «Файл не найден!» |
| 1.3 Строка символов должна быть корректной | Если строка символов некорректна или не существует, сообщение: «Файл некорректен» | <строки нет> | Сообщение: «Файл некорректен» |
| 1. Преобразование строки source в destination | | | |
| 2.1 Строка преобразована | Если строка преобразована, в выходной файл выводится преобразованная строка. | 100  asjhd24-=,.ahaj | Выходной файл: «24-=,.» |

1. **Код программы**

#include <iostream>

#include <fstream>

#include <cctype>

#include <locale>

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

{

int counter = 0;

for (int i = 0; source[i] != '\0'; i++)

if (!isalpha(source[i]))

{

destination[counter] = source[i];

counter++;

}

destination[counter] = '\0';

return destination;

}

int main()

{

setlocale(LC\_ALL, "russian");

//работа с файлами

std::ifstream fileIn;

std::ofstream fileOut;

std::string fileNameIn = "in.txt";

std::string fileNameOut = "out.txt";

fileIn.open(fileNameIn);

if (!fileIn)

{

std::cerr << "Файл не существует!" << std::endl;

return -1;

}

fileOut.open(fileNameOut);

bool flag = true;

while (flag)

{

int length = 0;

char\* firstLine = new char[100];

fileIn.getline(firstLine, 100);

try {

length = atoi(firstLine);

}

catch (std::exception& err) {

std::cerr << "Файл имеет некорректные данные!" << std::endl;

delete[] firstLine;

return -1;

}

if (length <= 0)

{

std::cerr << "Длина строки - положительное число!" << std::endl;

delete[] firstLine;

return -1;

}

else

{

std::cout << "LEN: " << length << std::endl;

}

char\* secondLine = new char[length + 1];

char\* destinationString = new char[length + 1];

/\*if (!fileIn.getline(secondLine, length + 1))

{

std::cerr << "Файл некорректен" << std::endl;

delete[] firstLine;

delete[] secondLine;

delete[] destinationString;

return -1;

}\*/

fileIn.getline(secondLine, length + 1);

std::cout << secondLine << std::endl;

deleteLatin(destinationString, secondLine);

fileOut << "Полученная строка: " << destinationString << std::endl;

//следующая строка

if (fileIn.eof())

flag = false;

fileIn.clear();

if (strlen(secondLine) < length)

{

while (strlen(secondLine) > length -1 && fileIn.get() != '\n')

{

if (fileIn.eof()) {

flag = false;

break;

}

}

}

fileIn.clear();

delete[] firstLine;

delete[] secondLine;

delete[] destinationString;

}

fileIn.close();

fileOut.close();

return 0;

}