**Лабораторная работа 7**

по дисциплине «Алгоритмизация и программирование»

Выполнил студент Гамин В.И.

гр. 3530202/20002

Руководитель [Червинский](https://ruz.spbstu.ru/teachers/999300) А.П.

доцент, к.т.н.

**Строки в стиле C. Вариант 5.**

**Общая постановка задачи**

Имеются две строки, которые читаются из файла.

Необходимо создать новую строку из символов двух исходных строк, которые не являются для них общими (с помощью функции) и вывести ее в другой файл.

**Таблица с детальными требованиями**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Требование** | **Детальные требования** | **Данные** | **Ожидаемый результат** |  |
| 1. Файл для ввода должен существовать и быть успешно открытым | Если файл не смог открыться, то сообщение: “The file input is deleted or does not exist, program termination.” | input.txt | Сообщение: “The file input is deleted or does not exist, program termination.” |  |
| 2. Файл для вывода должен существовать и быть успешно открытым | Если файл не смог открыться, то сообщение: “The file input is deleted or does not exist, program termination.” | result.txt | Сообщение: “The file input is deleted or does not exist, program termination.” |  |
| 3. Первая строчка файла ввода должна содержать максимальную длину из двух входящих строк. | | | | |
| а) Первая строчка – не дробное число | a) Если первая строчка – дробное число, то сообщение: “The maximum length of the string must be set as a positive integer, program termination.” | 2.5  !@#$123  456@#$ | Сообщение: “The maximum length of the string must be set as a positive integer, program termination.” |  |
| б) Первая строчка – положительное число | б) Если первая строчка – число, меньшее или равное нулю, то сообщение: “The maximum length of the string must be set as a positive integer, program termination.” | -3  QWERT  12345TW | Сообщение: “The maximum length of the string must be set as a positive integer, program termination.” |  |
| в) Первая строчка - число | в) Если первая строчка – не число, то сообщение: “The maximum length of the string must be set as a positive integer, program termination.” | GOOD  GFDERT  GFTUJ4% | Сообщение: “The maximum length of the string must be set as a positive integer, program termination.” |  |

**Программа**

*#include* <iostream>

*#include* <fstream>

*#include* <string>

*char\** func(*char\** destination, *const* *char\** source1, *const* *char\** source2);

*int* main()

{

   std::ifstream in; *//поток для чтения*

   std::ifstream outin; *//*

   std::ofstream out; *// поток для записи*

   std::string path\_in = "";

   std::string path\_out = "";

   std::cout << "Enter the path of the input file: ";

   std::cin >> path\_in;

   in.open(path\_in);

*try* {

*if* (!in.is\_open()) {

*throw* "The file input is deleted or does not exist, program termination.\n";

      }

   }

*catch* (*const* *char*\* error) {

      std::cerr << error << "\n";

      exit(-1);

   }

   std::cout << "File input opened successfully\n";

*int* MAX\_SIZE = 0;

*try* {

      in >> MAX\_SIZE;

*if* (in.peek() == '.' || in.peek() == ',' || !in.good() || MAX\_SIZE <= 0) {

*throw* "The maximum length of the string must be set as a positive integer, program termination.\n";

      }

   }

*catch* (*const* *char*\* error) {

      std::cerr << error << "\n";

      exit(-1);

   }

   MAX\_SIZE = MAX\_SIZE \* 2 + 1;

   std::cout << "Enter the path of the output file: ";

   std::cin >> path\_out;

   outin.open(path\_out);

*try* {

*if* (!outin.is\_open()) {

*throw* "The file output is deleted or does not exist, program termination.\n";

      }

   }

*catch* (*const* *char*\* error) {

      std::cerr << error << "\n";

      exit(-1);

   }

   out.open(path\_out);

   std::cout << "File output opened successfully\n";

*char*\* string1 = *new* *char*[MAX\_SIZE];

*char*\* string2 = *new* *char*[MAX\_SIZE];

*char*\* answer = *new* *char*[MAX\_SIZE];

   in >> string1;

   in >> string2;

   std::cout << "Result: " << func(answer, string1, string2) << "\n";

   out << "Result: " << func(answer, string1, string2);

   delete[] string1;

   delete[] string2;

   delete[] answer;

   in.close();

   out.close();

   outin.close();

*return* 0;

}

*char\** func(*char\** destination, *const* *char\** source1, *const* *char\** source2)

{

*int* num = 0;

*int* k = 0;

*int* i = 0;

*int* j = 0;

*while* (source1[i] != '\0')

   {

      j = 0;

      k = 0;

*while* (source2[j] != '\0')

      {

*if* (source1[i] == source2[j])

         {

            k += 1;

         }

         j += 1;

      }

*if* (k == 0)

      {

         destination[num] = source1[i];

         num = num + 1;

      }

      i += 1;

   }

   i = 0;

   j = 0;

   k = 0;

*while* (source2[i] != '\0')

   {

      j = 0;

      k = 0;

*while* (source1[j] != '\0')

      {

*if* (source2[i] == source1[j])

         {

            k += 1;

         }

         j += 1;

      }

*if* (k == 0)

      {

         destination[num] = source2[i];

         num += 1;

      }

      i += 1;

   }

   destination[num] = '\0';

*return* destination;

}