**МИНОБРНАУКИ РОССИИ**

**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**

**Кафедра САПР**

**ОТЧЕТ**

**по лабораторной работе №2**

**по дисциплине «Программирование»**

**Тема: Работа с текстом**

Студент гр. 2302 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Николаев В.Ю.

Преподаватель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Калмычков В.А.

Санкт-Петербург

2023 г.

# Исходная формулировка

Задан текст, содержащий произвольное количество строк, в которых отдельные слова могут разделяться одним или несколькими пробелами и знаками пунктуации (переносы слов с одной строки на другую не используются). Сформировать новый текст, который является результатом одного из следующих преобразований исходного текста: каждое слово, встречающееся более одного раза в тексте, заменить начиная со второго вхождения на обозначение вида: S(NN), где S - первый знак слова; NN - порядковый номер в тексте первого вхождения слова.

# **Математическая постановка задачи**

Используемые условия:

C1: fin >> c && c != '\n'

C2: data == nullptr

C3: !fin.is\_open()

C4: fin >> c

C5: !data[i].Input(fin)

C6: j1 < data[i1].size

C7: !IsLetter(data[i1].data[j1])

C8: (i2 != i1 && j2 < data[i2].size) || (i2 == i1 && j2 < j1)

C9: !IsLetter(data[i2].data[j2])

C10: word1 == word2

C11: !flag

C12: !f1

# Контрольный пример

| Исходный текст | Полученный текст |
| --- | --- |
| 1. I wish I loved the Human Race; 2. I wish I loved the Human Race; 3. And when I'm Introduced to one, 4. I wish I loved its silly face; 5. I wish I liked the way it walks; 6. I wish I liked the way it talks; 7. And when I'm Introduced to one, 8. I wish I thought "What Jolly Fun!" | 1. I wish I(1) loved the Human Race; 2. I(1) w(2) I(1) l(4) t(5) H(6) R(7); 3. And when I(1)'m Introduced to one, 4. I(1) w(2) I(1) l(4) its silly face; 5. I(1) w(2) I(1) liked t(5) way it walks; 6. I(1) w(2) I(1) l(32) t(5) w(34) i(35) talks; 7. A(15) w(16) I(1)'m(18) I(19) t(20) o(21), 8. I(1) w(2) I(1) thought "What Jolly Fun!" |

# **Организация UI**

Макет E1:

Error. File Input.txt does not open.\n

Макет E2:

Error. Memory not allocated.\n

Макет E3:

Error. File Output.txt does not open.\n

| Макет файла input.txt:   1. c\_c c\_c … c\_c 2. c\_c c\_c … c\_c 3. ⋮ ⋮ ⋱ ⋮ 4. c\_c c\_c … c\_c | Макет файла output.txt:   1. c\_c c\_c … c\_c 2. c\_c c\_c … c\_c 3. ⋮ ⋮ ⋱ ⋮ 4. c\_c c\_c … c\_c |
| --- | --- |

# Структуры данных

Для решения поставленной задачи используются две структуры данных:

| Строка | Текст |
| --- | --- |
| class String  {  public:  char \*data;  int size = 0;  public:  String();  String(int);  ~String();  bool Input(std::fstream &);  void Output(std::fstream &);  friend bool operator==(const String &, const String &);  }; | class Text  {  public:  String \*data;  int size = 0;  bool haveErorrs = false;  private:  bool IsLetter(char);  char GetSmallLetter(char);  String GetWord(int, int);  void SetSizes(Text &);  void SetData(Text &);  public:  ~Text();  void Input();  Text Processing();  void Output();  }; |
| char \*data - указатель на массив символов строки;  int size - размер массива символов. | String \*data - указатель на массив строк;  int size - размер массива строк;  bool haveErorrs - если в результате исполнения программы произошла какая-либо ошибка, то данный флаг принимает значение true и программа завершается. |
|  |  |

# **Способ хранения данных**

| Тип данных | Название переменной | Описание переменной |
| --- | --- | --- |
| Text | start | Исходный текст |
| Text | end | Исходный текст после обработки |
| fstream | fin | Поток ввода из файла input.txt |
| fstream | fout | Поток вывода в файл output.txt |

# Описание функций

| Тип | Название | Передаваемые переменные | Описание переменных | Описание функции |
| --- | --- | --- | --- | --- |
| void | Text::  Input | - | - | Ввод текста из файла input.txt |
| Text | Processing | - | - | Обработка текста |
| void | Text::  Output | - | - | Вывод текста в файл output.txt |
| bool | IsLetter | char c | символ | Проверяет является ли символ буквой |
| char | GetSmallLetter | char c | буква | Возвращает прописную версию букву |
| String | GetWord | int i | номер первого элемента слова | Возвращает слово из текста, расположенное между первым и последним символом |
| int j | номер последнего элемента слова |
| void | SetSizes | Text &end | текст после обработки | Устанавливает размер текста после обработки |
| void | SetData | Text &end | текст после обработки | Устанавливает текст после обработки |
| bool | String::  Input | - | - | Ввод строки из файла input.txt |
| void | String::  Output | - | - | Вывод строки в файл output.txt |
| bool | operator  == | const String &left | левая строка | Проверяет равны ли строки |
| const String &right | правая строка |

# **Представление алгоритма решения задачи**

# 

# Текст программы

| src/   | Main.cpp | | --- | | #include "../lib/Text/Text.h"  int main()  {  Text start;  start.Input();  Text end = start.Processing();  end.Output();  return 0;  } |   lib/   | Header.h | | --- | | #pragma once  #include <iostream>  #include <fstream> |   lib/String/   | String.h | | --- | | #pragma once  #include "../Header.h"  class String  {  public:  char \*data;  int size = 0;    public:  String();  String(int);    ~String();  bool Input(std::fstream &);  void Output(std::fstream &);  friend bool operator==(const String &, const String &);  }; | | Constructors.cpp | | #include "String.h"  String::String() {}  String::String(int number)  {  int t = number;  while (t)  {  size++;  t /= 10;  }  data = new char[size];  t = size;  while (number)  {  data[--t] = number%10 + '0';  number /= 10;  }  } | | Destructor.cpp | | #include "String.h"  String::~String()  {  if (data != nullptr)  delete []data;  } | | Input.cpp | | #include "String.h"  bool String::Input(std::fstream &fin)  {  char c;  while (fin >> c && c != '\n')  size++;  size++;  for (int i = 0; i < size; i++)  fin.unget();  data = new char[size];  if (data == nullptr)  {  std::cout << "E1\n";  return false;  }  for (int i = 0; i < size; i++)  fin >> data[i];  return true;  } | | Operator.cpp | | #include "String.h"  bool operator==(const String &left, const String &right)  {  if (left.size == right.size)  for (int i = 0; i < left.size; i++)  if (left.data[i] != right.data[i])  return false;  return left.size == right.size;  } | | Output.cpp | | #include "String.h"  void String::Output(std::fstream &fout)  {  for (int i = 0; i < size; i++)  fout << data[i];  } |   lib/Text/   | Processing.cpp | | --- | | #include "Text.h"  Text Text::Processing()  {  Text end;  if (haveErorrs)  {  end.haveErorrs = true;  return end;  }  SetSizes(end);  SetData(end);  return end;  } | | SetData.cpp | | #include "Text.h"  void Text::SetData(Text &end)  {  for (int i1 = 0; i1 < size; i1++)  {  for (int j1 = 0, j = 0; j1 < data[i1].size;)  {  if (!IsLetter(data[i1].data[j1]))  {  end.data[i1].data[j] = data[i1].data[j1];  j1++, j++;  continue;  }  String word1 = GetWord(i1, j1);  int counter = 0;  bool f1 = false;  for (int i2 = 0; i2 <= i1; i2++)  {  bool flag = true;  for (int j2 = 0; (i2 != i1 && j2 < data[i2].size) || (i2 == i1 && j2 < j1); j2++)  {  if (!IsLetter(data[i2].data[j2]))  continue;  String word2 = GetWord(i2, j2);  counter++;  if (word1 == word2)  {  flag = false;  f1 = true;  break;  }  j2 += word2.size - 1;  }  if (!flag)  break;  }  if (!f1)  for (int k = 0; k < word1.size; k++)  end.data[i1].data[j++] = data[i1].data[j1 + k];  else  {  end.data[i1].data[j++] = data[i1].data[j1];  end.data[i1].data[j++] = '(';  String number(counter);  for (int k = 0; k < number.size; k++)  end.data[i1].data[j++] = number.data[k];  end.data[i1].data[j++] = ')';  }  j1 += word1.size;  }  } } | | lib/Text/   | Text.h | | --- | | #pragma once  #include "../String/String.h"  class Text  {  public:  String \*data;  int size = 0;  bool haveErorrs = false;  private:  bool IsLetter(char);  char GetSmallLetter(char);  String GetWord(int, int);    void SetSizes(Text &);  void SetData(Text &);  public:  ~Text();  void Input();  Text Processing();  void Output();  }; | | Destructor.cpp | | #include "Text.h"  Text::~Text()  {  if (data != nullptr)  delete []data;  } | | GetSmallLetter.cpp | | #include "Text.h"  char Text::GetSmallLetter(char c)  {  return ('A' <= c && c <= 'Z') ? c - 'A' + 'a' : c;  } | | GetWord.cpp | | #include "Text.h"  String Text::GetWord(int i, int j)  {  int k = j + 1;  while (k < data[i].size && IsLetter(data[i].data[k]))  k++;    int length = k - j;  String word;  word.size = length;  word.data = new char[size];  for (; j < k; j++)  word.data[j - k + length] = GetSmallLetter(data[i].data[j]);  return word;  } | | Input.cpp | | #include "Text.h"  void Text::Input()  {  std::fstream fin("text/Input.txt", std::ios::in);  if (!fin.is\_open())  {  std::cout << "E2\n";  fin.close();    haveErorrs = true;  return;  }  fin << std::noskipws;  char c;  while (fin >> c)  size += c == '\n';  size += c != '\n';  fin.close();  fin.open("text/Input.txt", std::ios::in);  if (!fin.is\_open())  {  std::cout << "E2\n";  fin.close();    haveErorrs = true;  return;  }  fin << std::noskipws;  data = new String[size];  for (int i = 0; i < size; i++)  if (!data[i].Input(fin))  {  haveErorrs = true;  return;  }  fin.close();  return;  } | | IsLetter.cpp | | #include "Text.h"  bool Text::IsLetter(char c)  {  return ('a' <= c && c <= 'z') || ('A' <= c && c <= 'Z');  } | | Output.cpp | | #include "Text.h"  void Text::Output()  {  if (haveErorrs)  return;  std::fstream fout("text/Output.txt", std::ios::out);  if (!fout.is\_open())  {  std::cout << "E3\n";  fout.close();  haveErorrs = true;  return;  }  for (int i = 0; i < size; i++)  data[i].Output(fout);  fout.close();  return;  } | | SetSizes.cpp | | #include "Text.h"  void Text::SetSizes(Text &end)  {  end.size = size;  end.data = new String[size];  for (int i1 = 0; i1 < size; i1++)  {  for (int j1 = 0; j1 < data[i1].size; j1++)  {  if (!IsLetter(data[i1].data[j1]))  {  end.data[i1].size++;  continue;  }  String word1 = GetWord(i1, j1);  int counter = 0;  bool f1 = false;  for (int i2 = 0; i2 <= i1; i2++)  {  bool flag = true;  for (int j2 = 0; (i2 != i1 && j2 < data[i2].size) || (i2 == i1 && j2 < j1); j2++)  {  if (!IsLetter(data[i2].data[j2]))  continue;    String word2 = GetWord(i2, j2);  counter++;  if (word1 == word2)  {  f1 = true;  flag = false;  break;  }  j2 += word2.size - 1;  }  if (!flag)  break;  }  end.data[i1].size += !f1 ? word1.size : 3 + String(counter).size;  j1 += word1.size - 1;  }  end.data[i1].data = new char[end.data[i1].size];  }  } | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |

# **Результаты работы программы**

Контрольный пример программа прошла успешно. Все некорректные ситуации программа обрабатывает корректно (сообщает о полученной ошибке).

# Выводы о проделанной работе

В этой лабораторной работе мы приобрели навыки работы с текстом.