|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | |  |  |  | | МИНОБРНАУКИ РОССИИ | | | | Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА – Российский технологический университет»**  **РТУ МИРЭА** | | |   Институт Информационных технологий | |
|  | |
| Кафедра Математического обеспечения и стандартизации информационных технологий | |
|  | |
|  | |

|  |  |
| --- | --- |
| **ОТЧЕТ ПО ПРАКТИЧЕСКОЙ РАБОТЕ № 3** | |
| **по дисциплине** | |
| **«**Структуры и алгоритмы обработки данных**»**  **Тема: «Строки»** | |
|  | |
| Выполнил студент группы ИКБО-07-22 | Козлов К.И. |
| Принял преподаватель | Филатов А.С. |

|  |  |  |
| --- | --- | --- |
| Лабораторная работа выполнена | «\_\_»\_\_\_\_\_\_\_202\_\_ г. | *(подпись студента)* |
|  |  |  |
| «Зачтено» | «\_\_»\_\_\_\_\_\_\_202\_\_ г. | *(подпись руководителя)* |

Москва 2023

# **Цель работы**

- Получение навыков в разработке алгоритмов обработки текста – извлечение отдельных элементов.

- Получение навыков использования средств языка Си и С++ для реализации алгоритмов обработки текстовых данных.

# **Постановка задачи**

* 1. Разработать программу согласно задаче варианта, используя для представления обрабатываемого в программе текста нуль терминальную строку и средства языка С для выполнения операций над этой строкой.
  2. Разработать программу согласно задаче варианта, используя для представления обрабатываемого в программе текста строку string стандартной библиотеки шаблонов и возможности класса для выполнения действий со строкой.
  3. Тестирование программы должно производиться на введенных вручную данных.
  4. Составить отчет, отобразив в нем описание выполнения всех этапов разработки, тестирования и код всей программы со скриншотами результатов тестирования.

Требования к упражнениям:

Вариант №9. Условие задания:

|  |  |
| --- | --- |
| Упражнение 1 | Дано предложение и управляющий символ, значением которого может быть один из символов Y, N. Вывести этот текст без входящих в него цифр, если значение символа управления = Y ,а если этот символ = N, то перенести все цифры текста в конец предложения так, чтобы первая встреченная в исходном тексте цифра была последней цифрой, а последняя первой и был сохранен порядок следования остальных цифр. |

# **Решение**

В данной практической работе была проведена работа с разными видами строк. Был использован динамический массив символов и класс string.

Строковым типом называют тип данных, каждое значение которого представляет собой последовательность символов некоторого алфавита.

Значение строкового типа называют строкой. В данной работе строка представляет собой динамический массив символов, оканчивающийся символом \0 (Си строки).

Для такого типа строк память выделяется как для динамического массива с помощью функций malloc() и realloc().

Также для ввода и вывод строк используются стандартные потоки ввода и вывода cin, cout соответственно и методы стандартного потока ввода cin. Напрмре getline(), который имеет 3 аргумента: куда записывается информация(строка), какой максимальный размер можеть иметь и символ окончания строки.

Класс string предназначен для работы со строками типа char\*, которые представляют собой строку с завершающим нулем. Класс string был введенн как альтернативный вариант для работы со строками типа char\*.

Чтобы использовать возможности класса string в MS Visual Studio (C++), нужно подключить библиотеку <string> и пространство имен std.

Объявление переменной типа string осуществляется точно так же как и обычной переменной. Возможный вариант объявления с одновременной инициализацией.

Класс string был придуман, чтоб исключить недостатки Си строк. Так, например, в отличие от Си строк к string применимы стандартные операторы C++.

Для решения упражнения было написано несколько функций: DeleteNumsChar(), ShiftNumsChar(), DeleteNumsString(), ShiftNumsString(). Данные функции реализуют удаление и смещение цифр в строке, с которой они работают.

Функция DeleteNumsChar() предназначена для работы с Си строками и имеет один аргумент - динамический массив символов. Эта функция реализует удаление цифр из строки путем вынесения их из строки.

|  |
| --- |
| void DeleteNumsChar(char\* Sentence)  {  int i = 1;  int length = strlen(Sentence);  while (i < length)  {  if (((int)Sentence[i] <= 57) && ((int)Sentence[i] >= 48))  {  for (int j = i; j < length; j++)  {  Sentence[j] = Sentence[j + 1];  }  }  else  {  i++;  }  }  } |

Функция ShiftNumsChar() имеет один аргумент – динамический массив символов. Эта функция реализует сдвиг всех цифр в конец строки путем пузырьковой сортировки. Затем, эта функция меняет местами последний элемент строки и первую встреченную цифру в отсортированной строке. Таким образом не теряется порядок всех встреченных цифр в строке, лишь меняются местами первая и последняя цифры в строке.

|  |
| --- |
| void ShiftNumsChar(char\* Sentence)  {  int i = 1;  int length = strlen(Sentence);  for (int i = 1; i < length; i++)  {  for (int j = 1; j < length - i; j++)  {  if ((((int)Sentence[j] <= 57) && ((int)Sentence[j] >= 48)) && not(((int)Sentence[j + 1] <= 57) && ((int)Sentence[j + 1] >= 48)))  {  swap(Sentence[j], Sentence[j + 1]);  }  }  }  for (int i = 1; i < length; i++)  {  if (((int)Sentence[i] <= 57) && ((int)Sentence[i] >= 48))  {  swap(Sentence[i], Sentence[length - 1]);  break;  }  }  } |

Функция DeleteNumsString() имеет один аргумент – строку string. Она выполняет такой же функционал, как и DeleteNumsChar(), только предназначена для типа string и возвращает измененную строку.

|  |
| --- |
| string DeleteNumsString(string Sentence)  {  int i = 1;  int length = Sentence.length();  while (i < length)  {  if (((int)Sentence[i] <= 57) && ((int)Sentence[i] >= 48))  {  for (int j = i; j < length; j++)  {  Sentence[j] = Sentence[j + 1];  }  }  else  {  i++;  }  }  return (Sentence);  } |

Функция ShiftNumsString()имеет один аргумент – строку string. Она выполняет такой же функционал, как и функция DeleteNumsString(), только предназначена для типа string и возвращает измененную строку.

|  |
| --- |
| {  for (int i = 1; i < Sentence.length() - 1; i++)  {  for (int j = 1; j < Sentence.length() - 1; j++)  {  if ((((int)Sentence[j] <= 57) && ((int)Sentence[j] >= 48)) && not(((int)Sentence[j + 1] <= 57) && ((int)Sentence[j + 1] >= 48)))  {  swap(Sentence[j], Sentence[j + 1]);  }  }  }  for (int j = 1; j < Sentence.length(); j++)  {  if ((((int)Sentence[j] <= 57) && ((int)Sentence[j] >= 48)))  {  swap(Sentence[j], Sentence[Sentence.length() - 1]);  break;  }  }  return (Sentence);  } |

# **4. Тестирование**

Протестируем программу. Для этого выберем работу с динамическим массивом, нажав 1, введем строку и выберем режим удаления всех цифр.

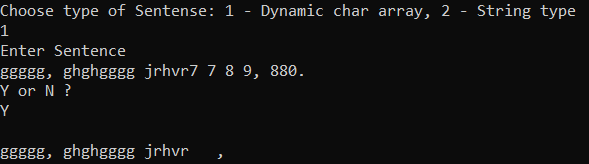


Рисунок 1. Удаление цифр из строки

Как видно, программа корректно удаляет все встреченные в строке цифры и оставляет все остальные символы.

Протестируем работу с типом string.2

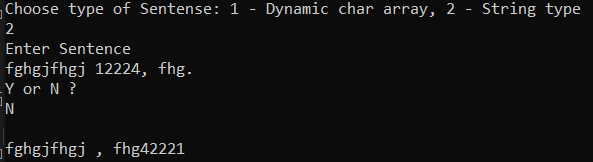


Рисунок 2. Смещение цифр в строке

Как мы видим, программа переносит все цифры в конец строки и меняет местами первую и последнюю встреченную в строке.

# **5. Выводы**

В результате выполнения работы я:

1. Освоены навыки в разработке алгоритмов обработки текста – извлечение отдельных элементов
2. Освоены навыки использования средств языка Си и С++ для реализации алгоритмов обработки текстовых данных.

# **6. Исходный код программы**

|  |
| --- |
| #if defined(\_WIN32) || defined(\_\_WIN32\_\_) || defined(WIN32)  #define \_CRT\_SECURE\_NO\_WARNINGS  #pragma warning(disable:4996)  #endif  #include <iostream>  #include <string>  using namespace std;  void DeleteNumsChar(char\* Sentence)  {  int i = 1;  int length = strlen(Sentence);  while (i < length)  {  if (((int)Sentence[i] <= 57) && ((int)Sentence[i] >= 48))  {  for (int j = i; j < length; j++)  {  Sentence[j] = Sentence[j + 1];  }  }  else  {  i++;  }  }  }  void ShiftNumsChar(char\* Sentence)  {  int i = 1;  int length = strlen(Sentence);  for (int i = 1; i < length; i++)  {  for (int j = 1; j < length - i; j++)  {  if ((((int)Sentence[j] <= 57) && ((int)Sentence[j] >= 48)) && not(((int)Sentence[j + 1] <= 57) && ((int)Sentence[j + 1] >= 48)))  {  swap(Sentence[j], Sentence[j + 1]);  }  }  }  for (int i = 1; i < length; i++)  {  if (((int)Sentence[i] <= 57) && ((int)Sentence[i] >= 48))  {  swap(Sentence[i], Sentence[length - 1]);  break;  }  }  }  string DeleteNumsString(string Sentence)  {  int i = 1;  int length = Sentence.length();  while (i < length)  {  if (((int)Sentence[i] <= 57) && ((int)Sentence[i] >= 48))  {  for (int j = i; j < length; j++)  {  Sentence[j] = Sentence[j + 1];  }  }  else  {  i++;  }  }  return (Sentence);  }  string ShiftNumsString(string Sentence)  {  for (int i = 1; i < Sentence.length() - 1; i++)  {  for (int j = 1; j < Sentence.length() - 1; j++)  {  if ((((int)Sentence[j] <= 57) && ((int)Sentence[j] >= 48)) && not(((int)Sentence[j + 1] <= 57) && ((int)Sentence[j + 1] >= 48)))  {  swap(Sentence[j], Sentence[j + 1]);  }  }  }  for (int j = 1; j < Sentence.length(); j++)  {  if ((((int)Sentence[j] <= 57) && ((int)Sentence[j] >= 48)))  {  swap(Sentence[j], Sentence[Sentence.length() - 1]);  break;  }  }  return (Sentence);  }  int main()  {  int Choise;  do  {  do  {  cout << "Choose type of Sentense: 1 - Dynamic char array, 2 - String type" << endl;  cin >> Choise;  } while (Choise != 1 && Choise != 2);  switch (Choise)  {  case 1:  {  char\* Sentence = (char\*)malloc(sizeof(char) \* 256);  char Control;  cout << "Enter Sentence" << endl;  cin.getline(Sentence, 256, '.');  cout << "Y or N ?" << endl;  cin >> Control;  if (Control == 'Y')  {  DeleteNumsChar(Sentence);  cout << Sentence << endl;  }  else  {  ShiftNumsChar(Sentence);  cout << Sentence << endl;  }  }  break;  case 2:  {  char Control;  string Sentence;  cout << "Enter Sentence" << endl;  getline(cin, Sentence, '.');  cout << "Y or N ?" << endl;  cin >> Control;  if (Control == 'Y')  {  cout << DeleteNumsString(Sentence) << endl;  }  else if (Control == 'N')  {  cout << ShiftNumsString(Sentence) << endl;  }  }  }  } while (Choise == 1 || Choise == 2);  } |