Министерство образование Республики Беларусь

Учреждение образования «Полоцкий государственный университет»

Информационно-технологический факультет

Кафедра технологий программирования

Лабораторная работа №1

По дисциплина «Основы алгоритмизации и программирования»

Выполнил: Крутилин С.Д.

Группа 19-ИТ-1.

Проверил: Адамовский Е.

Новополоцк

2019 г.

**Строки**

**Вариант 2**

**Цель работы**: научиться работать с векторными данными языка «С++». Изучить правила объявления строки и обращения к её элементам, а также библиотечные функции для работы со строками. Освоить основные алгоритмы обработки строк.

**Задание 1. Вывести первый, последний и средний (если он есть)) символы строки.**

Для решения это задачи необходимы функции, которые будут считывать строку и искать ее длину. По условию для чтения нужно было воспользоваться функцией gets() из библиотеки <cstdio>, но по официальным данным самая последняя редакция стандарта C (2011) окончательно удалила эту функцию из своей спецификации. Эта функция устарела в C ++ (начиная со стандарта 2011 года, который следует за C99 + TC3). Вместо нее я воспользовался манипулятором потока get() из библиотеки <iostream>.

#include <iostream>

#include <cstring>

using namespace std;

int main()

{

setlocale(LC\_ALL, "ru");

const int SIZE = 256;

char str[SIZE] = "";

cout << "Введите строку: ";

cin.get(str, SIZE);

cout << "Первый символ строки: ";

cout << str[0];

cout << endl;

cout << "Последний символ строки: ";

cout << str[strlen(str) - 1];

cout << endl;

if (strlen(str) % 2 == 1) {

cout << "Средний символ строки: ";

cout << str[strlen(str) / 2];

}

else {

cout << "Средний символ отсутствует.";

}

}

**Задание 2. Даны две строки. Определить, содержится ли меньшая по длине строка в большей.**

Для решения этой задачи необходимо считать две строки, сравнить их длину, чтобы определить большую строку, а затем проверить на наличие меньшей строки в большей. По условию мы можем использовать только библиотеки <cstdlib> и <cstring>, функциями gets() и puts() для вывода. Функции gets() больше не поддерживается, поэтому я ее заменил манипулятором потока get().

#include <iostream>

#include <cstring>

#include <cstdlib>

using namespace std;

int main()

{

setlocale(LC\_ALL, "ru");

const int SIZE = 256;

char strFirst[SIZE] = "";

cout << "Введите строку: ";

cin.get(strFirst, SIZE);

cin.get();

char strSecond[SIZE] = "";

cout << "Введите вторую строку: ";

cin.get(strSecond, SIZE);

if (strlen(strFirst) > strlen(strSecond)) {

if (strstr(strFirst, strSecond)) {

cout << "Меньшая строка:\t\t\t\t";

puts(strSecond);

cout << "Содержиться в большей строке:\t\t";

puts(strFirst);

}

else {

cout << "Меньшая строка:\t\t\t\t";

puts(strSecond);

cout << "Не содержиться в большей строке:\t\t";

puts(strFirst);

}

}

else {

if (strstr(strSecond, strFirst)) {

cout << "Меньшая строка:\t\t\t\t";

puts(strFirst);

cout << "Содержиться в большей строке:\t\t";

puts(strSecond);

}

else {

cout << "Меньшая строка:\t\t\t\t";

puts(strFirst);

cout << "Не содержиться в большей строке:\t\t";

puts(strSecond);

}

}

cout << "\n\n\n\n\n\n\n\n\n\n\n\n\n";

}

**Задание 3. Удалить в строке все символы '!'.**

Краткий алгоритм решения: чтение строки символов, удаления символов, вывод новой строки. В языке нет возможности просто удалить символ из массива без использования сторонних библиотек, поэтому мы просто перезапишем нашу строку в новую без восклицательных знаков.

#include <iostream>

#include <cstring>

#include <cstdlib>

using namespace std;

int main()

{

setlocale(LC\_ALL, "ru");

const int SIZE = 256;

char str[SIZE] = "";

char strRe[SIZE] = "";

cout << "Введите строку: ";

cin.get(str, SIZE);

cin.get();

for (int j = 0, i = 0; i < strlen(str); i++, j++) {

if (str[i] == '!') {

continue;

}

strRe[j] = str[i];

}

cout << "Строка без восклицательных знаков: ";

puts(strRe);

cout << "\n\n\n\n\n\n\n\n\n\n\n\n\n";

}

**Задание 4. Найти символ в строке, встречающийся наибольшее число раз.**

Краткий алгоритм решения данной задачи: считать строку, перебрать каждый элемент массива и найти встречающийся наибольшее число раз.  
Считывание: в этой задачи я использую gets\_s(насколько я понял эта функция существует с 11-ого стандарта, т.е. после того, как удалили get).   
Перебор: цикл for.  
Поиск встречающегося наибольшее число раз элементов: используя счетчик пробегаем по всей строке(ищем сколько наш элемент встречался в строке раз), а затем сравниваем значения счетчика со значением max. При необходимости перезаписываем в max новое значение.

#include <iostream>

using namespace std;

int main() {

setlocale(LC\_ALL, "ru");

const int size = 256;

char str[size], maxs = '\0';

cout << "Введите строку: " << endl;

gets\_s(str);

int max = 0;

for (int i = 0; str[i]; i++) {

int n = 0;

for (int j = i; str[j]; j++)

n += str[i] == str[j];

if (max < n) {

max = n;

maxs = str[i];

}

}

cout << "Ваша строка: " << endl;

puts(str);

cout << "Символ в строке, встречающийся наибольшее число раз: ";

cout << maxs << endl;

return 0;

}

**Задание 5. Дана строка, содержащая слова. Вывести на экран слово, являющееся последним в алфавитном порядке.**

Краткий алгоритм решения задачи: считывание строки, добавить в конец стоки пробел для удобства обработки слов, перебрать слова и после каждого сделать проверку, какой из них ниже в алфавитном порядке, перезаписать наше последнее слово при необходимости, сделать вывод нашего слова.  
Для чтения использован манипулятор потока get() из библиотеки <iostream> по причине отсутствия функции gets() в современных стандартах языка.   
Для добавления в конец строки использована strcat\_s().Компилятор снова ругался на strcat() и рекомендовал заменить функцию на более безопасную.  
Перебор осуществляем в цикле for и в каждой итерации выполняем проверку на конец слова, а затем и на положения в списке. При необходимости перезаписываем наше последнее слово.  
Вывод с помощь рекомендованного puts().

#include <iostream>

#include <cstdlib>

#include <cstring>

using namespace std;

int main()

{

const int SIZE = 256;

char str[SIZE]="", strcurrent[SIZE] = "", strlast[SIZE] = "";

int k = 0, i;

cin.get(str,SIZE);

strcat\_s(str, " ");

for (i = 0; i < strlen(str); i++) {

if (str[i] != ' ')

{

strcurrent[k] = str[i];

strcurrent[k + 1] = '\0';

k++;

}

else

{

if (strcmp(strcurrent,strlast) == 1) {

for (int j = 0; j < strlen(strcurrent); j++) {

strlast[j] = strcurrent[j];

}

}

strcurrent[0] = '\0';

k = 0;

}

}

puts(strlast);

return 0;

}