Белорусский государственный технологический университет

Факультет информационных технологий

Кафедра программной инженерии

 Лабораторная работа 13

По дисциплине «Основы алгоритмизации и программирования»

На тему «Обработка символьной информации»

Выполнил:

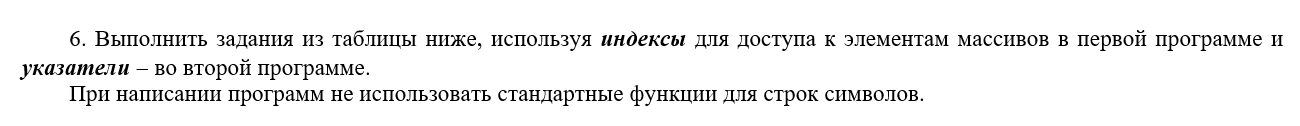
Студент 1 курса 6 группы

Романов Игорь Вячеславович

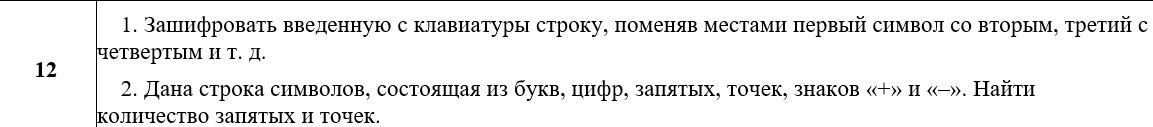
Преподаватель: асс. Андронова М.В.

2023, Минск

Задание 6



Вариант 12



1.

#include <iostream>

#include <cstring>

using namespace std;

//функция для шифрования строки

void encodeString(char str[], int size) {

int temp = 0; //инициализация временной переменной

//если размер кратен 2, то поменять местами все цифры

if (size % 2 == 0) {

for (int i = 0; i < size; i += 2) {

temp = str[i];

str[i] = str[i + 1];

str[i + 1] = temp;

}

cout << "Строка после шифрования: " << str << endl; //вывод строки после шифрования

}

//если размер не кратен 2, то поменять местами все символы, кроме последнего, а его просто вывести

else {

for (int i = 0; i < size - 1; i += 2) {

temp = str[i];

str[i] = str[i + 1];

str[i + 1] = temp;

}

cout << "Строка после шифрования: " << str << endl; //вывод строки после шифрования

}

}

//основная функция

int main() {

setlocale(LC\_CTYPE, "Russian");

int size = 0; //ввод размера строки

char str[256]; //ввод строки

cout << "Введите размер строки: "; cin >> size;

cout << "\nВведите строку: ";

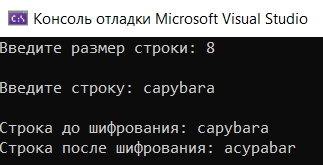
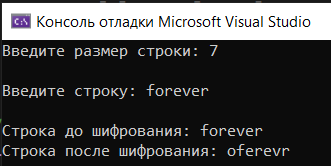
cin >> str; //чтобы считывать всю строку, включая пробелы

cout << "\nСтрока до шифрования: " << str << endl;

encodeString(str, size);

return 0;

}



2.

#include <iostream>

using namespace std;

//функция для подсчета запятых и точек в строке

void countCommasDots(char str[], int size) {

int comma\_counter = 0, dot\_counter = 0; //инициализация счетчиков

for (int i = 0; i < size; i++) {

if (\*(str + i) == ',') { //в случае нахождения запятой в строке увеличить соответствующий счетчик

comma\_counter++;

}

if (\*(str + i) == '.') { //в случае нахождения точки в строке увеличить соотвествующий счетчик

dot\_counter++;

}

}

cout << "\nКоличество запятых в строке: " << comma\_counter << endl; //вывод количества запятых в строке

cout << "Количество точек в строке: " << dot\_counter << endl; //вывод количества точек в строке

}

//основная функция

int main()

{

setlocale(LC\_CTYPE, "Russian");

int size = 0; //ввод размера строки

char str[256]; //ввод строки

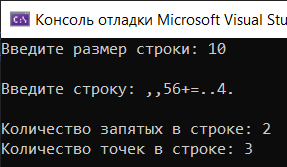
cout << "Введите размер строки: "; cin >> size;

cout << "\nВведите строку: "; cin >> str;

countCommasDots(str, size); //подсчет запятых и точек

return 0;

}



Доп.задания

Вариант 1



#include <iostream>

#include <cstring>

using namespace std;

//функция для сравнения символа с окончанием слова

void getEnding(char str[], char ending, int& position, int& i, bool& flag) {

while (str[i] != '\0') { //пока строка не закончится

if (str[i] != ' ' && flag == false) { //поиск начала слова и его фиксация

flag = true;

position = i;

}

else if ((str[i] == ' ' || str[i + 1] == '\0') && flag == true) {

bool hasEnding = (str[i] == ending); // проверяем, заканчивается ли слово на указанный символ

if (hasEnding) {

for (int j = position; j <= i; j++) {

cout << str[j];

}

cout << " ";

}

flag = false; //указываем, что слово закончилось

while (str[i] == ' ') {

i++; // пропускаем пробелы

}

}

i++;

}

}

//основная функция

int main() {

setlocale(LC\_CTYPE, "Russian");

char str[256], ending; //ввод строки и проверяемого окончания

int position = 0, i = 0; //ввод позиции проверяемого конца слова и порядка букв в строке

bool flag = false; //ввод флажка, сообщающего о начале слова

cout << "Введите строку: \n";

cin.getline(str, 256);

cout << "Введите проверяемое окончание: \n";

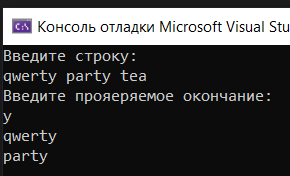
cin >> ending;

cin.ignore(); // очистка буфера после ввода символа

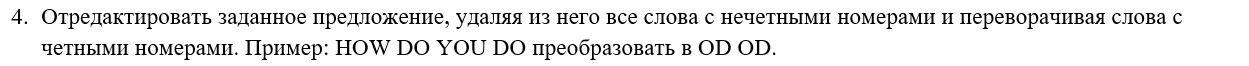
getEnding(str, ending, position, i, flag);

return 0;

}



Вариант 4



#include <iostream>

#include <cstring>

using namespace std;

//функция для редактирования строки в зависимости от порядка

void editString(char str[], char str\_copy[]) {

int index = 1, i = 0; //инициализация переменных

while (str[i] != '\0') { //пока строка не закончится

if (str[i] != ' ' && index % 2 != 0) { //если слово нечетного порядка

while (str[i] != ' ' && str[i] != '\0') {

i++;

}

index++;

}

if (str[i] != ' ' && index % 2 == 0) { //если слово четного порядка

char temp[256]{};

int k = 0;

while (str[i] != ' ' && str[i] != '\0') {

temp[k] = str[i];

k++;

i++;

}

temp[k] = '\0';

\_strrev(temp);

strcat\_s(str\_copy, 256, temp); // реверс и копирование символов в str\_copy

strcat\_s(str\_copy, 256, " "); // добавление пробела между словами

index++;

}

i++;

}

cout << str\_copy; // вывод полученной строки str\_copy

}

//основная функция

int main() {

setlocale(LC\_CTYPE, "Russian");

char str[256], str\_copy[256]{}; //ввод строки и ее копии

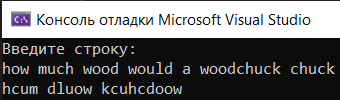
cout << "Введите строку: \n";

cin.getline(str, 256);

editString(str, str\_copy); //переход к функции

return 0;

}



Вариант 3



#include <iostream>

#include <cstring>

using namespace std;

//функция для редактирования строки в зависимости от порядка

void removeRepeats(char str[]) {

int i = 0, check = 0; //инициализация переменных

char str\_copy[256]{}; //инициализация копии строки

while (str[i] != '\0') { //пока строка не закончится

if (str[i] == ' ') { //пропускам пробелы в строке

i++;

}

if (str[i] != ' ') { //если встретился не пробел

char temp[256]{};

int k = 0;

while (str[i] != '\0' && str[i] != ' ') { //временной переменной присвается новое слово

temp[k] = str[i];

k++;

i++;

}

temp[k] = '\0';

check = strspn(temp, str\_copy); //проверяется, входит ли слово из temp в строку str\_copy

if (!check) { //если нет, то присвоить строке данное слово

strcat\_s(str\_copy, 256, temp);

strcat\_s(str\_copy, 256, " "); // добавление пробела между словами

}

}

}

cout << str\_copy; // вывод полученной строки str\_copy

}

//основная функция

int main() {

setlocale(LC\_CTYPE, "Russian");

char str[256]; //ввод строки и ее копии

cout << "Введите строку: \n";

cin.getline(str, 256);

removeRepeats(str); //переход к функции

return 0;

}

