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:

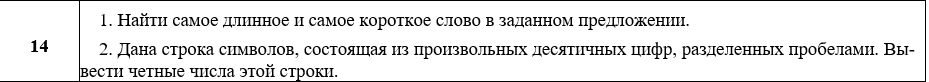
Студент 1 курса 10 группы

Сегренёв Кирилл Сергеевич

Преподаватель: асс. Андронова М.В.

2023, Минск

**ВАРИАНТ 14**



**Задание 1.**

**Код:**

#include <iostream>

#include <vector>

#include <string>

using namespace std;

int main() {

setlocale(LC\_ALL, "ru");

char str[] = "dfgsdfg sgdgf dgdfg dg dfg";

int k = 0, j = 0, max = 0;

for (int i = 0; str[i] != '\0'; i++) { // Создаем цикл для вычисления количества слов в предложении

// Делаем это для дальнейшей записи данных слов в массив

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

k++;

}

}

k++;

vector<string> words(k);

for (int i = 0; i < k; i++) { // Записываем слова в новый массив

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

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

j++;

}

j++;

}

// Создаем новые массивы для дальнейшего сравнения слов

string shortestWord = words[0];

string longestWord = words[0];

for (const auto& word : words) { // В этом цикле мы сравниваем слова по количеству символов

// Данный цикл будет проходить по всем словам пока не дойдет до последнего

if (word.length() < shortestWord.length()) {

shortestWord = word;

}

if (word.length() > longestWord.length()) {

longestWord = word;

}

}

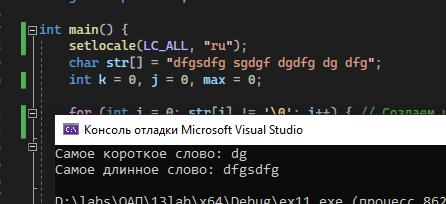
cout << "Самое короткое слово: " << shortestWord << endl;

cout << "Самое длинное слово: " << longestWord << endl;

return 0;

}

**Output:**

****

**Задание 2.**

**Код:**

#include <iostream>

using namespace std;

int main() {

char str[] = "12 21 54 234 45 6 123 34 6 4653 6"; // Инициализируем массив

int k = 0, j = 0;

for (int i = 0; str[i] != '\0'; i++) { // Считаем сколько чисел будет в массиве

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

k++;

}

}

k++;

int\* numbers = new int[k];

for (int i = 0; i < k; i++) { // Записываем числа в новый массив уже типа int для дальнейших арифметических операций

\*(numbers + i) = 0;

while (\*(str + j) != ' ' && \*(str + j) != '\0') {

\*(numbers + i) = \*(numbers + i) \* 10 + (\*(str + j) - '0');

j++;

}

j++;

}

for (int i = 0; i < k; i++) { // Выводим те числа, которые кратны 2

if(\*(numbers + i) % 2 == 0)

{

cout << \*(numbers + i) << " ";

}

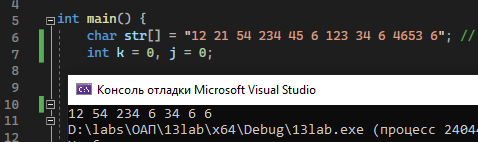
}

delete[] numbers;

return 0;

}

**Output:**



**ДОПОЛНИТЕЛЬНЫЕ ЗАДАНИЯ**

**Задание 1**

В заданной последовательности слов найти все слова, имеющие заданное окончание.

**Код:**

#include <iostream>

#include <string>

using namespace std;

// 1 доп

int main() {

setlocale(LC\_ALL, "ru");

const int maxSize = 300;

int k = 0;

char arr[maxSize];

char symbol;

cout << "Введите последовательность слов через пробел(до 300 символов): " << endl;

cin.getline(arr, maxSize); // Используем функци getline для считывания символов(считая пробелы) с клавиатуры. Функцию сразу же записывает данные значения в массив arr

cout << "Введите окончание: " << endl;

cin >> symbol;

int r = strlen(arr); // Вычисляем длину нашего массива (данная функция считает только символы, которые заняты нами в памяти, а те, которые не введенные, не считает

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

k++; // Длина слова

if (arr[i] == ' ' || arr[i] == '\0') {

k--;

if (arr[i - 1] == symbol) {

for (int g = i - k; arr[g] != '\0' && arr[g] != ' '; g++) {

cout << arr[g];

}

cout << ' ';

}

k = 0;

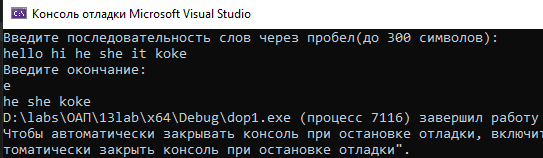
}

}

return 0;

}

**Output:**



**Задание 4**

Отредактировать заданное предложение, удаляя из него все слова с нечетными номерами и переворачивая слова с четными номерами. Пример: HOW DO YOU DO преобразовать в OD OD

**Код:**

#include <iostream>

#include <string>

using namespace std;

// 4 доп

int main() {

setlocale(LC\_ALL, "ru");

const int maxSize = 300;

char arr[maxSize];

cout << "Введите предложение: " << endl;

cin.getline(arr,maxSize); // Вводим предложение и сразу записываем его в массив

int k = 0;

int r = strlen(arr); // Узнаем длину нашего массива с символами

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

if (arr[i] == ' ' || arr[i] == '\0') {

k++; // Добавляем k для более удобного отслеживания четных элементов массива

if (k % 2 == 0) { // Если число кратно 2, то выводить его символы в другом порядке

for (int g = i - 1; arr[g] != '\0' && arr[g] != ' '; g--) {

cout << arr[g];

}

cout << ' ';

}

}

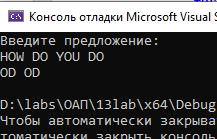
}

cout << '\n';

return 0;

}

**Output:**

****

**Задание 3**

Из заданного предложения удалить те слова, которые уже встречались в предложении раньше.

**Код:**

#include <iostream>

#include <vector>

using namespace std;

int main() {

char arr[] = "Hello he he she it he he hi Hello";

int k = 0;

// Подсчет количества слов в предложении

for (int i = 0; arr[i] != '\0'; i++) {

if (arr[i] == ' ') {

k++;

}

}

k++;

// Создаем массив для хранения слов

string\* words = new string[k];

// Записываем в этот массив наши слова

int j = 0;

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

while (arr[j] != ' ' && arr[j] != '\0') {

words[i] = words[i] + arr[j];

j++;

}

j++;

}

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

bool rep = 0; // Отслеживание, повторяется ли слово

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

if (words[i] == words[g]) { // Если слова одинаковые, то присвоить нашему индикатору число и выйти из цикла

rep = 1;

break;

}

}

if (rep == 0) { // Если слово не повторялось, то вывести его

cout << words[i] << ' ';

}

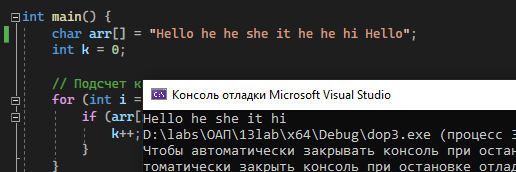
}

delete[] words;

return 0;

}

**Output:**

****