**Правительство Российской Федерации**

**Федеральное государственное автономное образовательное учреждение высшего профессионального образования   
"Национальный исследовательский университет   
"Высшая школа экономики"**

Московский институт электроники и математики Национального

исследовательского университета "Высшая школа экономики"

Департамент прикладной математики

**ОТЧЕТ**

**По лабораторной работе № 7**

**По курсу «Алгоритмизация и программирование»**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| |  |  | | --- | --- | |  | ФИО студента | | Номер группы | Дата |
| Колодин Матвей Алексеевич | БПМ213 | 01.02.2022 |
|  |
|  |
|  |

**Москва – 2022 г.**

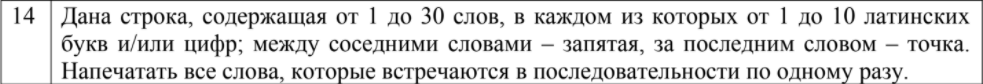
**ЗАДАНИЕ:**

**Написать функцию обработки строки и программу, тестирующую данную**

**функцию. В программе должен быть предусмотрен вывод исходной строки,**

**которая при выделении слов не должна измениться.**

**(вариант №14)**



**РЕШЕНИЕ:**

#include <stdio.h>

#include <malloc.h>

#include <stdlib.h>

#include <string.h>

int main(){

// Ввод.

int n = 0;

int lp = 0;

char st[360]; // 30 - максимум слов. 12 символов на слово, так как до 10 - слово, 1 - запятая / точка, 1 - ноль строчки).

printf("Input string:\n");

gets(st);

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

if(st[i] == ',' || st[i] == '.'){

++n;

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

if(st[j] >= 48 && st[j] <= 57){

}

else if(st[j] >= 65 && st[j] <= 90){

}

else if(st[j] >= 97 && st[j] <= 122){

}

else{

printf("Incorrect symbol - %c, pos - %d \n", st[j], j);

}

}

lp = i+1;

}

}

if(n > 30){

printf("Too many words, error(");

return 0;

}

else{

printf("Number of words - %d \n", n);

}

char a[n][12]; // массив строк a отвечает за разбиение исходной строки пословно. Делал задачу до знания strtok.

// Первичная обработка.

int pos=0, k=0, tmp=0;

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

if(st[i] == ','){

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

a[k][tmp] = st[j];

tmp++;

}

a[k][tmp] = '\0'; // - это очень важно при посимвольной инициализации строки. Конец строки)

k++;

tmp = 0;

pos = i + 1;

}

if(st[i] == '.'){

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

a[k][tmp] = st[j];

tmp++;

}

a[k][tmp] = '\0'; // - это очень важно при посимвольной инициализации строки. Конец строки)

break;

}

}

// Вычислительная часть.

int flg = -1;

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

for(int j = i+1; j < n; ++j){

if(strcmp(a[i],a[j]) == 0){

flg = 1;

a[j][0] = '\0'; // Выкидываем строчку, которая повторяется.

}

if(flg == 1 && j == n-1){

a[i][0] = '\0';

}

}

flg = -1;

}

// Вывод.

printf("Result:\n");

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

if(a[i][0]!='\0'){

printf("%s ",a[i]);

}

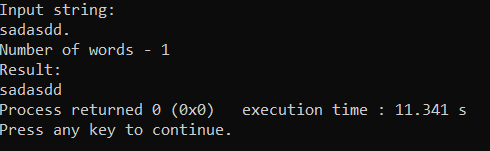
}

}

**ТЕСТЫ:**

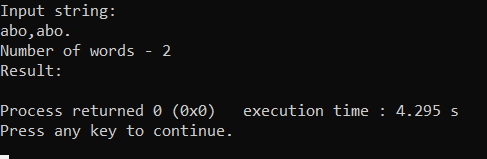
1. **Тесты идут в смешанном порядке – с результатом и без. Ко всем тестам есть пояснения.**
2. **Много слов + некорректный символ (будет предупреждение.)**

**Тест № 1**



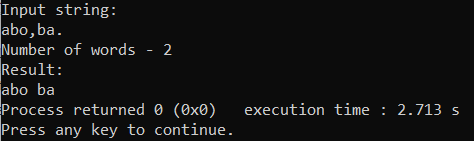
*(На ввод подается лишь одно слово.)*

**Тест № 2**



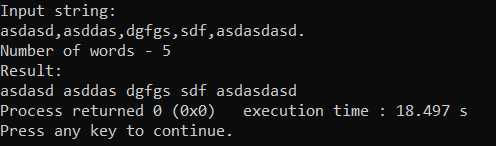
*(На ввод поданы два одинаковых слова. Результат – пустая строка.)*

**Тест № 3**



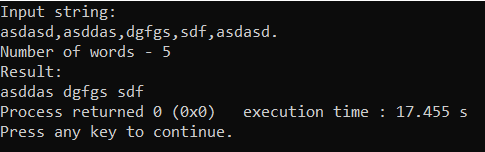
*(На ввод поданы два разных слова. Результат – два слова.)*

**Тест № 4**



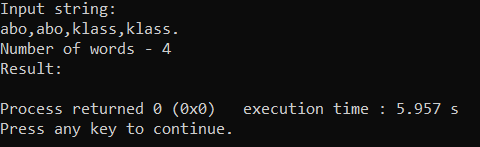
*(Чуть более сложный тест, увеличили количество слов.)*

**Тест № 5**



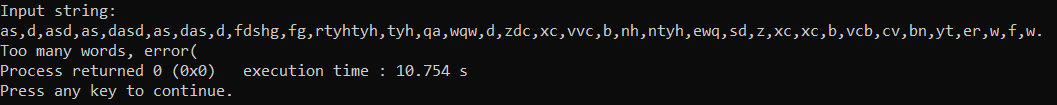
*(Есть повторяющееся слово – “asdasd”. Его мы и выкдиываем.)*

**Тест № 6**

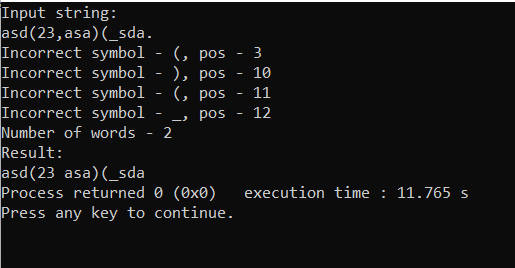


*(Все слова повторились. Результат – пустая строка.)*

**Тест № 7**

*(Количество слов больше 30.)*

**Тест № 8**



*(Некорректные символы, но в данном случае выполнение работы программы не прерывается. Есть только предупреждение.)*