

Балтийский государственный технический университет
«ВОЕНМЕХ» им. Д. Ф. Устинова

Кафедра И5 «Информационные системы и программная инженерия»

Практическая работа №6
по дисциплине «Информатика: Основы программирования»
на тему «Строки»

Выполнил:
Студент Альков В.С.
Группа И407Б

Преподаватель:
Першин Д.В.

Санкт-Петербург
2020 г.

Задача 1. Дана строка символов, содержащая предложение, заканчивающееся точкой. Найти в строке самое длинное слово и вывести его на экран.

Исходные данные:

str – строка, массив символов типа *char*.

Резльтирующие данные:

печать *str*.

Вспомогательные переменные:

i – индекс массива символов *str*, тип *int*.

mc – максимальная найденная длина слова, тип *int*.

p – указатель на слово, тип *char**.

Таблица тестирования:

This is a good day to die.	This is a good daaaay.day
This is a good day to die.	This is a good daaaay.
This	daaaay
Process returned 0 (0x0)	Process returned 0 (0x0)

```
This is a good day to die.This is a good day to die.
This is a good day to die.
This
Process returned 0 (0x0)   execution time : 12.715 s
```

```
This is a goood day.
This is a goood day.
goood
Process returned 0 (0x0)
```

Вспомогательные алгоритмы:

1) Алгоритм проверки символа на букву.

Входные данные: символ.

Резльтирующие данные: отсутствуют.

Этот алгоритм описывается в программе функцией

int check(char)

Параметры:

первый параметр – код символа.

Возвращаемое значение – 1, если переданный символ – буква, 0, если нет.

2) Алгоритм нахождения длины слова.

Входные данные: имя строки.

Резльтирующие данные: отсутствуют.

Этот алгоритм описывается в программе функцией

*int word_len(char *)*

Параметры:

первый параметр – адрес начала слова.

Возвращаемое значение – длина слова.

Вспомогательные переменные:

i – тип *int*, индекс переданного массива символов, переменная подсчета длина слова.

3) Алгоритм удаления символов строки после определенного символа.

Входные данные: имя строки, определенный символ.

Результирующие данные: удалены символы в строке после определенного символа.

Этот алгоритм описывается в программе функцией

*void cut_str(char *, char)*

Параметры:

первый параметр – адрес начала строки.

второй параметр – символ, после которого нужно удалить символы.

Возвращаемое значение – отсутствует.

Вспомогательные переменные:

i – тип *int*, индекс переданного массива символов.

Текст программы

```
#include <stdio.h>
#include <string.h>

/*объявление ф-ий*/
int check(char);
int word_len(char*);
void cut_str(char*, char);
int main()
{
    /*объявление переменных*/
    char str[80] = "", *p;
    int i, mc=0;
    /*ввод строки*/
    gets(str);
    /*удаление символов после точки*/
    cut_str(str, '.');
    /*вывод строки*/
    printf("%s\n", str);
    /*поиск слов в строке и подсчет их длины,
    нахождение слова максимальной длины*/
    for (i=0; str[i]!='\0' && str[i]!='.'; i++)
        if (check(str[i]) && !check(str[i-1]))
            if (word_len(&str[i])>mc)
            {
                p = &str[i];
                mc = word_len(&str[i]);
            };
    /*запись слова с максимальной длиной в str*/
```

```

        for(i=0; i<mc; i++)
            str[i] = *p++;
        str[i] = '\0';
        printf("%s",str);
        return 0;
    }
    /*Ф-ия проверки символа на букву*/
    int check(char a)
    {
        if ((a>='a' && a <='z') || (a>='A' && a <='Z'))
            return 1;
        return 0;
    }
    /*Ф-ия нахождения длины слова*/
    int word_len(char *a)
    {
        int i;
        for(i=0; check(*a); i++, a++);
        return i;
    }
    /*Ф-ия удаления символов в строке
    после определенного символа*/
    void cut_str(char *a, char b)
    {
        int i;
        for(i=0; a[i+1]!='\0'&&a[i]!=b; i++);
        a[i+1] = '\0';
    }

```

Задача 2. Дана строка, содержащая фамилию, имя и отчество человека. Преобразовать строку так, чтобы сначала указывались инициалы с точками, а затем фамилия.

Исходные данные:

str – строка, массив символов типа *char*

Результирующие данные:

печать str.

Вспомогательные переменные:

i – счетчик кол-ва сдвигов указателя *p* на длину слова+1, тип *int*.

str2 – темповая строка, массив символов типа *char*.

p – указатель на первую букву слова, тип *char**.

Таблица тестирования:

Alkov Valentin Sergeevich	Alkov Valentin Sergeevich Sergeevich
V. S. Alkov	V. S. Alkov
Process returned 0 (0x0)	Process returned 0 (0x0) execution

Вспомогательные алгоритмы:

- 1) Алгоритм проверки символа на букву и описывающая его функция те же, что и в предыдущей задаче

- 2) Алгоритм нахождения длины слова и описывающая его функция те же, что и в предыдущей задаче

Текст программы

```
#include <stdio.h>
#include <string.h>
/*объявление ф-ий*/
int check(char);
int word_len(char*);

int main()
{
    /*объявление переменных*/
    char str[80] = "", str2[80] = "", *p = str;
    int i;
    /*ввод строки*/
    gets(str);
    for(i=0; i<2; i++)
    {
        /*сдвигаем указатель на длину слова+1,
        чтобы он указывал на первую букву*/
        p += word_len(p) + 1;
        /*записываем в новую строку первую
        букву текущего слова с точкой*/
        sprintf(str2, "%s%c. ", str2, *p);
    };
    /*приписываем к str2 фамилию*/
    strncat(str2, str, word_len(str));
    /*заменяем старую строку новой*/
    strcpy(str, str2);
    /*выводим строку*/
    printf("%s", str);
    return 0;
}

/*ф-ия проверки символа на букву*/
int check(char a)
{
    if ((a>='a' && a<='z') || (a>='A' && a<='Z'))
        return 1;
    return 0;
}

/*ф-ия нахождения длины слова*/
int word_len(char *a)
{
    int i;
    for(i=0; check(*a); i++, a++);
    return i;
}
```