

Министерство образования и науки РФ

**НГТУ**

Кафедра Автоматики

**Лабораторная работа № 4**

*Указатели, функции*

**Вариант: 6**

**Факультет: АВТФ**

**Группа: АП-819**

**Студент: Зубарев Василий**

**Преподаватель:**

Саблина

Галина

Владимировна

**Дата: 28.04.09**

Новосибирск 2009

**Цель работы:** Освоить правила написания и использования функций в языке СИ. Научиться использовать указатели при обработке массивов данных.

**Задание:**

1. Написать программу сортировки массива строк по вариантам. Ввод данных, сортировку и вывод результатов оформить в виде функций. Входные и выходные параметры функции сортировки указаны в таблице. Входные и выходные параметры функций для ввода-вывода приведены в таблице.

2. Модифицировать программу п.1., применив в функциях передачу параметров и возврат результатов по ссылке (с использованием указателей). Сравнить результаты.

Задание	Входные данные	Выходные данные
Расположить строки по убыванию количества слов	1. Массив 2. Размерность массива	1. Количество перестановок 2. Максимальное количество слов

**Листинг программы:**

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#define MAX 5

int inp_str(int i, char *str) {
    printf("Enter string #d: ", i+1);
    gets(str);
    return(strlen(str));
}

int out_str(char *str, int len) {
    printf("\nString \"%s\" and it's length %d", str, len);
    return 0;
}

int space_count (char *str) {
    int i = 0, count = 0;

    while (str[i] != '\0') {
        if (str[i] == ' ') {
            count++;
        }
        i++;
    }
    return(count);
}
```

```
}
```

```
int sort (char **str, int *len) {
    int i, j, x, count = 0;
    char *pstr;

    for (i = 0; i < MAX-1; i++) {
        for (j = i+1; j < MAX; j++) {
            if (space_count(str[i]) < space_count(str[j]))
        {
                pstr = str[i];
                str[i] = str[j];
                str[j] = pstr;

                x = len[i];
                len[i] = len[j];
                len[j] = x;

                count++;
            }
        }
    }
    return count;
}
```

```
int main(int argc, char** argv) {
    int i, count, len[MAX];
    char *str[MAX];

    for (i = 0; i < MAX; i++) {
        str[i] = (char *)malloc(80);
        len[i] = inp_str(i, str[i]);
    }

    count = sort(str, len);
    printf("\nSorted array: ");
    for (i = 0; i < MAX; i++) {
        out_str(str[i], len[i]);
    }
    printf("\nMax words - %d", space_count(str[MAX - 1]) +
1);
    printf("\nReplaces - %d", count);

    free(str);
    return 0;
}
```

### **Пример вывода:**

```
Enter string #1: ddfd ddfd d
Enter string #2: ddfddfdiddfd
Enter string #3: sdsd sdsf gdgd sdfs
Enter string #4: sdsdsdss dsds
Enter string #5: dsfdfsfsfsssss
```

Sorted array:

```
String "sdsd sdsf gdgd sdfs" and it's length 19
String "dfd ddfd d" and it's length 11
String "sdsdsdss dsds" and it's length 13
String "dfddfdiddfd" and it's length 12
String "dsfdfsfsfsssss" and it's length 13
Max words - 1
Replaces - 3
```

**Вывод:** В ходе лабораторной работы мы научились круто использовать функции и даже передавать в нее параметры двумя способами — по ссылке и по значению.