

**Министр науки и высшего образования Российской
Федерации**

**Федеральное государственное автономное
образовательное учреждение высшего образования**

**«Национальный исследовательский университет
ИТМО»**

**Факультет информационных технологий и
программирования**

Лабораторная работа №8

Выполнил студент группы № М3104

Гурман Тимофей Владимирович

Подпись:



Проверил:

Повышев Владислав Вячеславович

Санкт-Петербург

2020

Текст задания

Задания

1. Осуществить конкатенацию (сложение) двух строк.
2. Осуществить конкатенация (сложение) первой строки и n начальных символов второй строки.
3. Осуществить сравнение двух строк.
4. Осуществить сравнение первых n символов двух строк.
5. Осуществить копирование одной строки в другую строку.
6. Осуществить копирование первых n символов одной строки в другую строку.
7. Определить длину строки.
8. Осуществить поиск в строке первого вхождения указанного символа.
9. Осуществить поиск в строке последнего вхождения указанного символа.
10. Осуществить поиск в одной строке любого из множества символов, входящих во вторую строку.
11. Определить длину отрезка одной строки, содержащего символы из множества символов, входящих во вторую строку.
12. Определить длину отрезка одной строки, не содержащего символы второй строки
13. Выделить из одной строки лексемы (кусочки), разделенные любым из множества символов (разделителей), входящих во вторую строку.

Решение с комментариями

//10 вариант

//задания 1,3,5,10,12

```
#include <stdio.h>
int Len(char*s) {
    int i = 0;
    while (s[i] != '\0') {
        i++;
    }
    return i;
}

void Sum(char*s1, char*s2) {
    char sum_res[200];
    for (int i = 0; i < Len(s1); i++) {
        sum_res[i] = s1[i];
    }
    for (int i = 0; i < Len(s2); i++) {
        sum_res[i + Len(s1)] = s2[i];
    }
    for (int i = 0; i < Len(s2) + Len(s1); i++) {
        printf("%c", sum_res[i]);
    }
}

int Comp(char*s1, char*s2) {
    int min;
    if (Len(s1) <= Len(s2)) {
        min = Len(s1);
    }
    else {
        min = Len(s2);
    }
    for (int i = 0; i < min; i++) {
        if (s1[i] > s2[i]) {
            return 1;
        } else if (s1[i] < s2[i]) {
            return -1;
        }
    }
    if (Len(s1) == Len(s2)) {
        return 0;
    } else if (Len(s1) > Len(s2)) {
        return 1;
    } else {
        return -1;
    }
}
}
```

```

void Copy(char*s1, char*s2) {
    for (int i = 0; i < Len(s1)+1; i++) {
        s2[i] = s1[i];
    }
    for (int i = 0; i < Len(s2); i++) {
        printf("%c", s2[i]);
    }
}

```

```

int Enter(char*s1, char*s2) {
    for (int i = 0; i < Len(s1); i++) {
        for (int j = 0; j < Len(s2); j++) {
            if (s1[i] == s2[j]) {
                return s1[i];
            }
        }
    }
    return '0';
}

```

```

int Segment(char*s1, char *s2) {
    int count = 0;
    int max_count = 0;
    for (int i = 0; i < Len(s1); i++) {
        for (int j = 0; j < Len(s2); j++) {
            if (s1[i] == s2[j]) {
                count = -1;
            }
        }
        count++;
        if (max_count < count) {
            max_count = count;
        }
    }
    return max_count;
}

```

```

int main() {
    char s1[100], s2[100], s1_copy[100];
    gets(s1);
    gets(s2);

    Sum(s1, s2); //1

    printf("\n");
    printf("%d\n", Comp(s1, s2)); //2

    Copy(s1, s1_copy); //3
}

```

```
printf("\n");  
printf("%c\n", Enter(s1, s2)); //4  
  
printf("%d", Segment(s1, s2)); //5  
  
return 0;  
}
```