

**Министерство 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 Вариант 11

Название работы: Работа со строками

Выполнил студент группы № М3113

Крамской Вадим Вадимович

Подпись:



**Санкт-Петербург
2022**

Условие:

1. Осуществить конкатенация (сложение) первой строки и n начальных символов второй строки.
2. Осуществить сравнение первых n символов двух строк.
3. Осуществить копирование первых n символов одной строки в другую строку.
4. Определить длину отрезка одной строки, содержащего символы из множества символов, входящих во вторую строку.
5. Выделить из одной строки лексемы (кусочки), разделенные любым из множества символов (разделителей), входящих во вторую строку.

Решение:

1)Была подключена библиотека <string.h>. Сложение первой строки и n символов второй было реализовано с помощью функции `strncat` .

```
//2
printf( format: "Zadanie 2\n");
int n;
scanf( format: "%d", &n);
strncat( Dest: str1, Source: str2, Count: n);
printf( format: "%s\n", str1);
```

2)Сравнение первых n символов строк было реализовано с помощью функции `memcmp`

```
//4
printf( format: "Zadanie 4\n");
scanf( format: "%d", &n);
int result = memcmp( Buf1: str1, Buf2: str2, Size: n);
1 if (result == 0) {
2     printf( format: "First %d symbols equal\n", n);
3 } else {
4     printf( format: "First %d symbols not equal\n", n);
5 }
6 }
```

3)копирование первых n символов одной строки в другую было реализовано с помощью функции `strncpy`, но сначала была создана

копия 1 строки с помощью функции strcpy.

```
//6
```

```
printf( format: "Zadanie 6\n");  
scanf( format: "%d", &n);  
char *strcpy;  
strcpy = malloc( Size: n1 * (sizeof(char)));  
strcpy( Dest: strcpy, Source: str1);  
strncpy( Dest: strcpy, Source: str2, Count: n);  
  
printf( format: "%s\n", strcpy);  
free( Memory: strcpy);
```

4) Данное задание было выполнено поэтапно: сортировка всех элементов 2 строки, после создание массива, в который каждый элемент строки входит 1 раз и проверка каждого символа из 1 строки.

```

//11
printf( format: "Zadanie 11\n");
int count = 1;
char *massel;
massel = malloc( Size: n2 * (sizeof(char)));
strcpy( Dest: massel, Source: str2);
bubblesort( mass: massel, n: n2);
char *str2sort;
str2sort = malloc( Size: n2 * sizeof(char));
str2sort[0] = massel[0];
} for (int i = 1; i < n2; ++i) {
}     if (massel[i] != massel[i - 1]) {
}         str2sort[count] = massel[i];
}         count += 1;
}     }
}
int proverk = 0;
int maxdl = 0;
int dl = 0;
} for (int i = 0; i < n1; ++i) {
}     proverk = (str1[i], str2sort, count);
}     if (proverk != 0) {
}         ++dl;
}     } else {
}         if (dl > maxdl) {
}             maxdl = dl;
}             dl = 0;
}         }
}     }
}
} if (dl > maxdl) {
}     maxdl = dl;
} }
printf( format: "%d\n", maxdl);
free( Memory: massel);

```

5) В данном задании использовался массив из 4 задания, шла проверка каждого символа из 1 строки, после сего выделялись необходимые кусочки.

```

//13
printf( format: "Zadanie 13\n");
proverk = 0;
int nomsim = 0;
char *kus;
kus = malloc( Size: n1 * sizeof(char));
int flag = 0;
for (int i = 0; i < n1; ++i) {
    proverk = prov( sim: str1[i], mass2: str2sort, n2: count);
    if (proverk != 0) {
        if (flag == 1) {
            for (int j = 0; j < nomsim; ++j) {
                printf( format: "%c", kus[j]);
            }
            printf( format: "\n");
            nomsim = 0;
        }
        flag = 1;
    } else {
        if (flag == 1) {
            kus[nomsim] = str1[i];
            nomsim += 1;
        }
    }
}
}

```

Вывод: в ходе лабораторной работы была изучена библиотека <string.h> и использованы несколько функций из нее, который позволили удобно выполнить поставленные задачи.