Министерство науки и высшего образования РФ
ФГАОУ ВО «Уральский федеральный университет
имени первого Президента России Б. Н. Ельцина»
ИРИТ-РТФ
Центр ускоренного обучения

# ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №4

по дисциплине «Прикладное программирование»

**Тема:** ***Приобретение навыков обработки строковых данных***

Студент группы РИЗ-200028у: И. С. Арсентьев
Преподаватель: О. Л. Чагаева,
ст. преподаватель

Екатеринбург 2022

**СОДЕРЖАНИЕ**

## 1 Постановка задачи

Цель:

1. Познакомиться с библиотечными функциями работы со строками, написать программу, которая использует эти функции.
2. Написать свои варианты варианты функций:
    a) Определения длины строки(3 способа)
    b) Копирования строк
    c) Сравнения строк
    d) Конкатенации строк
3. Переписать функции так, чтобы они использовали динамическую память при задании строк
4. Изменить программу так, чтобы вместо malloc() использовалась функция callloc(), найти сходство и различия этих функций
5. В программе организовать массив строк, применить функции, написанные в предыдущем пункте к строкам, составляющим этот массив

# 2 Описание работы

## 2.1. Выполнение поставленных задач.

**1.** ***Познакомиться с библиотечными функциями работы со строками, написать программу, которая использует эти функции.***

strlen() — определение дины строки;

strcat() — конкатенация строк;

strcpy() - копирование строк

strcmp() - сравнение строк

```
#include <string.h>
#include <iostream>
using namespace std;
void main() {
	setlocale(LC_ALL, "ru");
	char expl[20] = "12345656";
	char expl1[20] = "vcbcbcv";
	char copyArr[20];
	cout << "Исходная строка" << endl;
	cout << expl << endl;
	cout << "Длина строки" << endl;
	cout << strlen(expl) << endl;
	cout << "Конкатенация строк" << endl;
	strcat_s(expl, expl1);
	cout << expl << endl;
	cout << "Копирование строк" << endl;
	strcpy_s(copyArr, expl);
	cout << copyArr << endl;
	cout << "Сравнение строк" << endl;
	cout << strcmp(expl1, expl) << endl;
}
```

***2. Написать свои варианты варианты функций:***

- ***a) Определения длины строки (3 способа)***
- ***b) Копирования строк***
- ***c) Сравнения строк***
- ***d) Конкатенации строк***

```
#include <iostream>
using namespace std;
void main()
{
	//прототипы функций
	int dlina1(char*);
	int dlina2(char*);
	int dlina3(char*);
```

```
	void kopir(char*, char*);
	int compr(char*, char*);
	void concat(char*, char*);
	char str[] = "qwerty";
	char str1[25] = "1234567890";
	//вывод
	cout << dlina1(str) << endl;
	cout << dlina2(str) << endl;
	cout << dlina3(str) << endl;
	cout << str1 << endl;
	kopir(str1, str);
	cout << str1 << endl;
	cout << compr(str, str1) << endl;
	cout << str1 << endl;
	cout << str << endl;
	concat(str1, str);
	cout << str1 << endl;
}
//функция,определяющая длину строки с помощью цикла for
int dlina1(char* arr)
{
	int count = 0;
	for (int i = 0; arr[i] != '\0'; i++)
	{
		count++;
	}
	return count;
}
//функция,определяющая длину строки с помощью цикла while
int dlina2(char* arr)
{
	char* chr = arr;
	int count = 0;
	while (*chr != '\0') {
		count++;
```

```
          chr++;
     }
     return count;
}
//функция,определяющая длину строки с помощью перевода массива в
//строку и применения к ней стандартой функции length()
int dlina3(char* arr)
{
     string str(arr);
     return str.length();
}
//функция копирующая одну строку в другую
void kopir(char* arr1, char* arr)
{
     for (int i = 0; arr[i] != '\0'; i++)
     {
          arr1[i] = arr[i];
     }
}
//функция,сравнивающая строки
int compr(char* arr, char* arr1)
{
     int i = 0, result = 0;
     while (((arr[i] != '\0') && (arr1[i] != '\0')))
     {
          if (int(arr[i]) > int(arr1[i]))
          {
               result = 1;
               break;
          }
          if (int(arr[i]) < int(arr1[i]))
          {
               result = -1;
               break;
          }
```

```
          i++;
     }
     return result;
}
// функция,"склеивающая" строки (конкатенация)
void concat(char* a, char* b)
{
     int len = dlina2(a);
     len += 0;
     int i = 0;
     for (; b[i] != '\0'; i++)
     {
          a[len + i] = b[i];
     }
     a[len + i] = '\0';
}
```

***3. Написание функций так, чтобы они использовали динамическую память при создании строк.***

```
#include <iostream>
#include <cstdlib>
using namespace std;
void main()
{
     int dlina1(char*);
     int dlina2(char*);
     int dlina3(char*);
     void kopir(char*, char*);
     int compr(char*, char*);
     void concat(char*, char*);
     char* str, * str1;
     // Задаем динамически строки
     str = (char*)malloc(100);
     str1 = (char*)malloc(100);
     strncpy_s(str, 7, "asdad", 7);
     strncpy_s(str1, 11, "134536765", 11);

     cout << dlina1(str) << endl;
     cout << dlina2(str) << endl;
     cout << dlina3(str) << endl;
     cout << str1 << endl;
     kopir(str1, str);
     cout << str1 << endl;
     cout << compr(str, str1) << endl;
     cout << str1 << endl;
     cout << str << endl;
     concat(str1, str);
     cout << str1 << endl;
```

```
     free(str);
     free(str1);
}
int dlina1(char* arr)
{
     int count = 0;
     for (int i = 0; arr[i] != '\0'; i++)
     {
          count++;
     }
     return count;
}
int dlina2(char* arr)
{
     char* chr = (char*)malloc(100);
     chr = arr;
     int count = 0;
     while (*chr != '\0') {
          count++;
          chr++;
     }
     return count;
     free(chr);
}
int dlina3(char* arr)
{
     string str(arr);
     return str.length();
}
void kopir(char* arr1, char* arr)
{
     for (int i = 0; arr[i] != '\0'; i++)
     {
          arr1[i] = arr[i];
     }
}
int compr(char* arr, char* arr1)
{
     int i = 0, result = 0;
     while (((arr[i] != '\0') && (arr1[i] != '\0')))
     {
          if (int(arr[i]) > int(arr1[i]))
          {
               result = 1;
               break;
          }
          if (int(arr[i]) < int(arr1[i]))
          {
               result = -1;
               break;
          }
          i++;
     }
     return result;
}
void concat(char* a, char* b)
{
     int len = dlina2(a);
     len += 0;
```

```
    int i = 0;
    for (; b[i] != '\0'; i++)
    {
        a[len + i] = b[i];
    }
    a[len + i] = '\0';
}
```

4. ***Изменить программу так чтобы вместо функции mallcol() использовать calloc()***

```
#include <iostream>
#include <cstdlib>
using namespace std;
void main()
{
    int dlina1(char*);
    int dlina2(char*);
    int dlina3(char*);
    void kopir(char*, char*);
    int compr(char*, char*);
    void concat(char*, char*);
    char* str, * str1;
    //используем calloc() вместо malloc()
    str = (char*)calloc(100, sizeof(char));
    str1 = (char*)calloc(100, sizeof(char));
    strncpy_s(str, 7, "qwerty", 7);
    strncpy_s(str1, 11, "1234567890", 11);
    cout << dlina1(str) << endl;
    cout << dlina2(str) << endl;
    cout << dlina3(str) << endl;
    cout << str1 << endl;
    kopir(str1, str);
    cout << str1 << endl;
    cout << compr(str, str1) << endl;
    cout << str1 << endl;
    cout << str << endl;
    concat(str1, str);
    cout << str1 << endl;
    free(str);
    free(str1);
}
int dlina1(char* arr)
{
    int count = 0;
    for (int i = 0; arr[i] != '\0'; i++)
    {
        count++;
    }
    return count;
}
int dlina2(char* arr)
{
    char* chr = (char*)calloc(100, sizeof(char));
    chr = arr;
    int count = 0;
    while (*chr != '\0') {
        count++;
        chr++;
    }
```

```
   return count;
   free(chr);
}
int dlina3(char* arr)
{
   string str(arr);
   return str.length();
}
void kopir(char* arr1, char* arr)
{
   for (int i = 0; arr[i] != '\0'; i++)
   {
        arr1[i] = arr[i];
   }
}
int compr(char* arr, char* arr1)
{
   int i = 0, result = 0;
   while (((arr[i] != '\0') && (arr1[i] != '\0')))
   {
        if (int(arr[i]) > int(arr1[i]))
        {
             result = 1;
             break;
        }
        if (int(arr[i]) < int(arr1[i]))
        {
             result = -1;
             break;
        }
        i++;
   }
   return result;
}
void concat(char* a, char* b)
{
   int len = dlina2(a);
   len += 0;
   int i = 0;
   for (; b[i] != '\0'; i++)
   {
        a[len + i] = b[i];
   }
   a[len + i] = '\0';
}
```

5. ***Организовываем массив строк, применив функции, написанные в предыдущем пункте к строкам, составляющим этот массив.***

```
#include <iostream>
#include <cstdlib>
#include <string>
using namespace std;
void main()
{
	setlocale(LC_ALL, "ru");
	int dlina1(char*);
	int dlina2(char*);
	int dlina3(char*);
	void kopir(char*, char*);
	int compr(char*, char*);
	void concat(char*, char*);
	//вводим масссив строк
	char* date1[7];
	char* words[7];
	// определяем строки динамически и тут же заполняем их
	for (int i = 0; i < 7; i++)
	{
		date1[i] = (char*)malloc(100);
		words[i] = (char*)malloc(100);
		strncpy_s(date1[i], 25, "25", 25);
		strncpy_s(words[i], 7, "qwerty", 7);
	}

	cout << dlina1(date1[1]) << endl;
	cout << dlina2(words[0]) << endl;
	cout << dlina3(date1[2]) << endl;
	cout << date1[0] << endl;

	cout << "kopir" << endl;
	kopir(words[0], date1[0]);
	cout << words[0] << endl;
	cout << "compare" << endl;
	cout << compr(date1[0], words[0]) << endl;
	cout << "concat" << endl;
	concat(words[0], date1[0]);
	cout << words[0] << endl;
	// освобождаем память выделенную динамически
	for (int i = 0; i < 7; i++)
	{
		free(date1[i]);
		free(words[i]);
	}
}
int dlina1(char* arr)
{
	int count = 0;
	for (int i = 0; arr[i] != '\0'; i++)
	{
		count++;
	}
	return count;
}
int dlina2(char* arr)
{
```

```
     char* chr = (char*)calloc(100, sizeof(char));
     chr = arr;
     int count = 0;
     while (*chr != '\0') {
          count++;
          chr++;
     }
     return count;
     free(chr);
}
int dlina3(char* arr)
{
     string str(arr);
     return str.length();
}
void kopir(char* arr1, char* arr)
{
     for (int i = 0; arr[i] != '\0'; i++)
     {
          arr1[i] = arr[i];
     }
}
int compr(char* arr, char* arr1)
{
     int i = 0, result = 0;
     while (((arr[i] != '\0') && (arr1[i] != '\0')))
     {
          if (int(arr[i]) > int(arr1[i]))
          {
               result = 1;
               break;
          }
          if (int(arr[i]) < int(arr1[i]))
          {
               result = -1;
               break;
          }
          i++;
     }
     return result;
}
void concat(char* a, char* b)
{
     int len = dlina2(a);
     //len += 0;
     int i = 0;
     for (; b[i] != '\0'; i++)
     {
          a[len + i] = b[i];
     }
     a[len + i] = '\0';
}
```

# 3 Выводы по лабораторной работе

В ходе лабораторной работы познакомились с библиотечными функциями работы со строками, написли программу, которая использует эти функции.

Написаны свои варианты варианты функций: определения длины строки(3 способа), копирования строк, сравнения строк, конкатенации строк.

Также переписаны функции так, чтобы они использовали динамическую память при задании строк

Изменена программа так, чтобы вместо malloc() использовалась функция callloc().

В программе организован массив строк, применены функции, написанные в предыдущем пункте к строкам, составляющим этот массив