МИНИСТЕРСТВО ТРАНСПОРТА РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ АГЕНТСТВО ЖЕЛЕЗНОДОРОЖНОГО ТРАНСПОРТА

Государственное бюджетное образовательное учреждение

высшего образования

«ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ПУТЕЙ СООБЩЕНИЯ ИМПЕРАТОРА АЛЕКСАНДРА I»

Кафедра «ИНФОРМАЦИОННЫЕ И ВЫЧИСЛИТЕЛЬНЫЕ СИСТЕМЫ»

Дисциплина: «Программирование(C)»

ОТЧЕТ

по лабораторной работе № 1

Вариант *17*

Выполнил студент Шефнер А.

Факультета *АИТ*

Группы *ИВБ-211*

Санкт-Петербург

2023

**Постановка задачи**

Требуется написать свою собственную реализацию стандартных функций strlen, strcpy, strcat и strcmp двумя разными способами.

**Пояснения:**

Мои 8 функций находятся в файлах strfuncs.h (объявление) и strfuncs.c (реализация). Первый способ реализации – с помощью индексации, второй – с помощью указателей. В файлах tests.h и tests.c находятся тесты, которые для каждой стандартной функции выводят в консоль результат выполнения этой функции и двух моих.

**Код программы**

**main.c**

#include <stdio.h>  
#include "tests.h"  
  
int main() {  
 printf\_s("\nTesting 8 string functions.\n\n");  
 test\_all();  
 return 0;  
}

**tests.h**

void test\_strlen(char\* str);  
  
void test\_strcpy(char\* str);  
  
void test\_strcat(char\* destination, char\* source);  
  
void test\_strcmp(  
 const char \*str1, const char \*str2,  
 const char \*str3, const char \*str4,  
 const char \*str5, const char \*str6);  
  
void test\_all();

**tests.c**

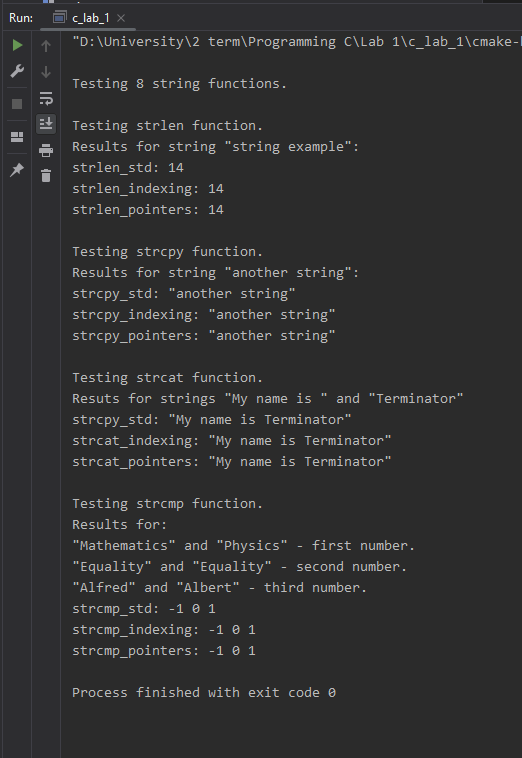
#include "string.h"  
#include "strfuncs.h"  
#include <stdio.h>  
#include "tests.h"  
  
void test\_strlen(char\* str) {  
 printf\_s("Testing strlen function.\n");  
  
 size\_t strlen\_std\_res = strlen(str);  
 size\_t strlen\_indexing\_res = strlen\_indexing(str);  
 size\_t strlen\_pointers\_res = strlen\_pointers(str);  
  
 printf\_s("Results for string \"%s\":\n", str);  
 printf\_s("strlen\_std: %d\n", strlen\_std\_res);  
 printf\_s("strlen\_indexing: %d\n", strlen\_indexing\_res);  
 printf\_s("strlen\_pointers: %d\n", strlen\_pointers\_res);  
 printf\_s("\n");  
}  
  
void test\_strcpy(char\* str) {  
 printf\_s("Testing strcpy function.\n");  
 printf\_s("Results for string \"%s\":\n", str);  
 size\_t len = strlen(str) + 1;  
 char \*strcpy\_std\_res[len];  
 char \*strcpy\_indexing\_res[len];  
 char \*strcpy\_pointers\_res[len];  
  
 strcpy(strcpy\_std\_res, str);  
 printf\_s("strcpy\_std: \"%s\"\n", strcpy\_std\_res);  
  
 strcpy\_indexing(strcpy\_indexing\_res, str);  
 printf\_s("strcpy\_indexing: \"%s\"\n", strcpy\_indexing\_res);  
  
 strcpy\_pointers(strcpy\_pointers\_res, str);  
 printf\_s("strcpy\_pointers: \"%s\"\n", strcpy\_pointers\_res);  
  
 printf\_s("\n");  
}  
  
void test\_strcat(char\* destination, char\* source) {  
 printf\_s("Testing strcat function.\n");  
 printf\_s("Resuts for strings \"%s\" and \"%s\"\n", destination, source);  
 size\_t source\_len = strlen(source) + 1;  
 size\_t dest\_len = strlen(destination) + source\_len + 1;  
  
 char\* strcat\_std\_dest[dest\_len];  
 strcpy(strcat\_std\_dest, destination);  
 char\* strcat\_indexing\_dest[dest\_len];  
 strcpy(strcat\_indexing\_dest, destination);  
 char\* strcat\_pointers\_dest[dest\_len];  
 strcpy(strcat\_pointers\_dest, destination);  
  
 strcat(strcat\_std\_dest, source);  
 printf\_s("strcpy\_std: \"%s\"\n", strcat\_std\_dest);  
  
 strcat\_indexing(strcat\_indexing\_dest, source);  
 printf\_s("strcat\_indexing: \"%s\"\n", strcat\_indexing\_dest);  
  
 strcat\_pointers(strcat\_pointers\_dest, source);  
 printf\_s("strcat\_pointers: \"%s\"\n", strcat\_pointers\_dest);  
  
 printf\_s("\n");  
}  
  
void test\_strcmp(  
 const char \*str1,  
 const char \*str2,  
 const char \*str3,  
 const char \*str4,  
 const char \*str5,  
 const char \*str6) {  
 printf\_s("Testing strcmp function.\n");  
 printf\_s("Results for:\n");  
 printf\_s("\"%s\" and \"%s\" - first number.\n", str1, str2);  
 printf\_s("\"%s\" and \"%s\" - second number.\n", str3, str4);  
 printf\_s("\"%s\" and \"%s\" - third number.\n", str5, str6);  
  
 int strcmp\_std\_res\_1 = strcmp(str1, str2);  
 int strcmp\_std\_res\_2 = strcmp(str3, str4);  
 int strcmp\_std\_res\_3 = strcmp(str5, str6);  
 printf\_s("strcmp\_std: %d %d %d\n", strcmp\_std\_res\_1, strcmp\_std\_res\_2, strcmp\_std\_res\_3);  
  
 int strcmp\_indexing\_res\_1 = strcmp\_indexing(str1, str2);  
 int strcmp\_indexing\_res\_2 = strcmp\_indexing(str3, str4);  
 int strcmp\_indexing\_res\_3 = strcmp\_indexing(str5, str6);  
 printf\_s("strcmp\_indexing: %d %d %d\n", strcmp\_indexing\_res\_1, strcmp\_indexing\_res\_2, strcmp\_indexing\_res\_3);  
  
 int strcmp\_pointers\_res\_1 = strcmp\_pointers(str1, str2);  
 int strcmp\_pointers\_res\_2 = strcmp\_pointers(str3, str4);  
 int strcmp\_pointers\_res\_3 = strcmp\_pointers(str5, str6);  
 printf\_s("strcmp\_pointers: %d %d %d\n", strcmp\_pointers\_res\_1, strcmp\_pointers\_res\_2, strcmp\_pointers\_res\_3);  
}  
  
void test\_all() {  
 test\_strlen("string example");  
 test\_strcpy("another string");  
 test\_strcat( "My name is ", "Terminator");  
 test\_strcmp("Mathematics","Physics",  
 "Equality","Equality",  
 "Alfred","Albert");  
}

**strfuncs.h**

#define size\_t unsigned long long  
  
// Indexing functions  
size\_t strlen\_indexing(const char \*str);  
  
char\* strcpy\_indexing(char \*destination, const char \*source);  
  
char\* strcat\_indexing(char \*destination, const char \*source);  
  
int strcmp\_indexing(const char \*str1, const char \*str2);  
  
// Pointer functions  
size\_t strlen\_pointers(const char \*str);  
  
char\* strcpy\_pointers(char \*destination, const char \*source);  
  
char\* strcat\_pointers(char \*destination, const char \*source);  
  
int strcmp\_pointers(const char \*str1, const char \*str2);

**strfuncs.c**

#include "strfuncs.h"  
#include "malloc.h"  
  
// Indexing functions  
size\_t strlen\_indexing(const char \*str) {  
 size\_t len = 0;  
 while (str[len] != '\0') {  
 len++;  
 }  
 return len;  
}  
  
char\* strcpy\_indexing(char \*destination, const char \*source) {  
 size\_t source\_len = strlen\_indexing(source) + 1;  
 for(int i = 0; i < source\_len; i++) {  
 destination[i] = source[i];  
 }  
 return destination;  
}  
  
char\* strcat\_indexing(char \*destination, const char \*source) {  
 size\_t dest\_len = strlen\_indexing(destination);  
 size\_t source\_len = strlen\_indexing(source) + 1;  
 for(int i = 0; i < source\_len; i++) {  
 destination[i + dest\_len] = source[i];  
 }  
 return destination;  
}  
  
int strcmp\_indexing(const char \*str1, const char \*str2) {  
 int cmp\_result;  
 int index = 0;  
 char str1\_tmp\_char, str2\_tmp\_char;  
 do {  
 str1\_tmp\_char = str1[index];  
 str2\_tmp\_char = str2[index];  
 cmp\_result = str1\_tmp\_char - str2\_tmp\_char;  
 if(cmp\_result != 0) break;  
 index++;  
 } while (str1\_tmp\_char != '\0' || str2\_tmp\_char != '\0');  
  
 if(cmp\_result < 0) return -1;  
 else if(cmp\_result > 0) return 1;  
 return 0;  
}  
  
// Pointer functions  
size\_t strlen\_pointers(const char \*str) {  
 const char\* str\_ptr = str;  
 while (\*str\_ptr != '\0') {  
 str\_ptr++;  
 }  
 return str\_ptr - str;  
}  
  
char \*strcpy\_pointers(char \*destination, const char \*source) {  
 char\* dest\_ptr = destination;  
 do {  
 \*(dest\_ptr++) = \*(source++);  
 } while (\*source != '\0');  
 return destination;  
}  
  
char \*strcat\_pointers(char \*destination, const char \*source) {  
 strcpy\_pointers(destination + strlen\_pointers(destination), source);  
 return destination;  
}  
  
int strcmp\_pointers(const char \*str1, const char \*str2) {  
 int cmp\_result;  
 while (\*str1 != '\0' || \*str2 != '\0') {  
 cmp\_result = \*(str1++) - \*(str2++);  
 if(cmp\_result != 0) break;  
 }  
  
 if(cmp\_result < 0) return -1;  
 else if(cmp\_result > 0) return 1;  
 return 0;  
}

**Отладка приложения:**

**Проверка в другом приложении (Mathcad)**