МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ, СВЯЗИ И МАССОВЫХ КОММУНИКАЦИЙ РОССИЙСКОЙ ФЕДЕРАЦИИ

Ордена Трудового Красного Знамени федеральное государственное бюджетное образовательное учреждение высшего образования

«МОСКОВСКИЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ СВЯЗИ И ИНФОРМАТИКИ»

(МТУСИ)

Кафедра «Информационная безопасность»

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №5

по дисциплине

«Программирование в системах информационной безопасности»

на тему

«Строки»

Вариант 26

Выполнил:

студент группы БСУ1801

Спирин В.В.

Проверил:

старший преподаватель кафедры ИБ

Барков В.В.

Москва, 2021

# Цель работы

Овладеть навыками работы с указателями и массивами в языке программирования C

# Задание

По номеру Вашего варианта выбрать задачу, решаемую в этой лабораторной работе, и выполнить для нее следующие задания.

**Общее Задание**

#### **Задание 1**

Написать функцию подсчёта дины строки. Функция принимает исходную строку и возвращает количество символов в строке, не включая нулевой символ.

Прототип функции int strlen(const char \*str).

Функция не должна использовать функции консольного ввода-вывода.

#### **Задание 2**

Написать функцию, поиска подстроки в строке. Функция принимает исходную строку и подстроку для поиска и возвращает адрес подстроки, если подстрока найдена, 0 в противном случае. Обращение к символам строки после нулевого символа недопустимо.

Прототип функции char \*find(const char \*str, const char \*substr).

Функция не должна использовать функции консольного ввода-вывода.

#### **Задание 3**

Написать функцию, удаления подстроки в строке. Функция принимает исходную строку и подстроку для поиска. С помощью функции find найти адрес начала подстроки и затем удалить эту подстроку. Оставшиеся в конце символы заполнить нулями. Обращение к символам строки после нулевого символа недопустимо.

Прототип функции void delete(char \*str, const char \*substr).

Функция не должна использовать функции консольного ввода-вывода.

#### **Задание 4**

Написать функцию копирования строк. Функция принимает исходную строку, буфер для копирования и размер буфера. Если длина исходной строки больше размера буфера, необходимо скопировать столько символов, сколько вместится в буфер, не забыв добавить в конце строки нулевой символ.

Прототип функции void strcpy(char \*dest, int len, const char \*src).

Функция не должна использовать функции консольного ввода-вывода.

#### **Задание 5**

Написать функцию конкатенации строк. Функция принимает исходные строки, буфер для размещения и размер буфера. Если сумма длин исходных строк больше размера буфера, необходимо скопировать столько символов, сколько вместится в буфер, не забыв добавить в конце строки нулевой символ.

Прототип функции void strcat(char \*dest, int len, const char \*first, const char \*second).

Функция не должна использовать функции консольного ввода-вывода.

#### **Задание 6**

Написать функцию вставки подстроки в строку. Функция принимает исходнную строку, строку для вставки, индекс, определяющий место вставки, буфер для размещения новой строки и размер буфера. Если сумма длин исходной строки и строки для встивки больше размера буфера или индекс находится за пределами строки, необходимо вернуть соответствующую ошибку.

Прототип функции \_Bool insert(const char \*src, const char \*str, int index, char \*dest, int len).

Функция не должна использовать функции консольного ввода-вывода.

#### **Задание 7**

Написать функцию main, демонстрирующую работу ранее разработанных функций

# Выполнение

Листинг 1 – программный код файла task1.c

int strlen(const char \*str)

{

int i = 0;

while (\*str)

{

++str;

++i;

}

return i;

}

Листинг 2 – программный код файла task2.c

char \*find(const char\* str, const char\* substr)

{

unsigned int i = 0;

unsigned int j, k;

while (str[i] != 0)

{

for (j = i, k = 0; substr[k] != 0 && str[j] == substr[k]; j++, k++)

{

if (substr[k+1] == 0)

{

return (str + i);

}

}

++i;

}

return 0;

}

Листинг 3 – программный код файла task3.c

#include <memory.h>

void delete(char \*str, const char \*substr)

{

int substrlen = strlen(substr);

char \*point = find(str, substr);

while (\*(point + substrlen))

{

\*(point) = \*(point + substrlen);

++point;

}

memset(point, 0, substrlen);

}

Листинг 4 – программный код файла task4.c

#include <memory.h>

void strcpy(char \*dest, int len, const char \*src)

{

memset(dest, 0, len);

char \*curDst = dest;

for (const char \*curSrc = src;

\*curSrc && curDst - dest <= len - 1;

++curSrc, ++curDst)

{

\*curDst = \*curSrc;

}

}

Листинг 5 – программный код файла task5.c

void strcat(char \*dest, int len, const char \*first, const char \*second)

{

strcpy(dest, len, first);

unsigned int firstLen = strlen(first);

unsigned int secondLen = strlen(second);

strcpy(dest + firstLen, secondLen, second);

}

Листинг 6 – программный код файла task6.c

\_Bool insert(const char \*src, const char \*str, int index, char \*dest, int len)

{

unsigned int strLenght = strlen(str);

unsigned int srcLen = strlen(src);

if (((strLenght + srcLen) > len) || (index > srcLen)) { return 0; }

char \*point = dest;

strcpy(point, index, src); //Копируем первую часть строки в буффер

point += index;

strcpy(point, strLenght, str); //Копируем подстроку в строку по индексу

point += strLenght;

strcpy(point, len, src + index); //Копируем оставшуюся часть строки в буффер

}

Листинг 7 – программный код файла l5.c

#include <stdio.h>

#include <stdbool.h>

#include <locale.h>

#include "Header.h"

void menu(int task\_count)

{

int i = 1;

while (i <= task\_count)

{

printf("%d. Задание %d\n", i, i);

++i;

}

printf("%d. Выход \n", i);

}

void main(void)

{

setlocale(LC\_ALL, "Rus");

\_Bool close\_flag = true;

while (close\_flag)

{

char str1[] = "Первая строка";

char substr[] = "стр";

char str3[] = " вторая ";

unsigned int len1 = 0, destlen = 0, index = 0;

\_Bool err;

char \*point , dest[100];

menu(6);

switch (getch())

{

case '1':

system("cls");

len1 = strlen(str1);

printf("Строка - %s. Длина строки - %d \n", str1, len1);

break;

case '2':

system("cls");

point = find(str1, substr);

point == 0 ? printf("Строка - %s. Подстрока - %s. \nПодстрока не входит в строку \n", str1, substr) : printf("Строка - %s. Подстрока - %s. \nАдрес вхождения - %p \n", str1, substr, point);

break;

case '3':

system("cls");

printf("Строка - %s. Подстрока - %s \n", str1, substr);

delete(str1, substr);

printf("Строка после удаления подстроки - %s \n", str1);

break;

case '4':

system("cls");

destlen = strlen(dest);

strcpy(dest, destlen, str1);

printf("Строка - %s. После копирования - %s \n", str1, dest);

break;

case '5':

system("cls");

destlen = strlen(dest);

strcat(dest, destlen, str1, str3);

printf("Строка - %s. Вторая строка - %s. \nРезультат конкатенации - %s \n", str1, str3, dest);

break;

case '6':

system("cls");

destlen = strlen(dest);

printf("Введите индекс для вставки ");

scanf\_s("%d", &index);

err = insert(str1, str3, index, dest, destlen);

err ? printf("Строка - %s. Вторая строка - %s. \nРезультат вставки - %s \n", str1, str3, dest) : printf("Ошибка \n");

break;

case '7':

close\_flag = false;

break;

}

}

}

Листинг 8 – программный код файла Header.h

int strlen(const char \*str);

char \*find(const char \*str, const char \*substr);

void delete(char \*str, const char \*substr);

void strcpy(char \*dest, int len, const char \*src);

void strcat(char \*dest, int len, const char \*first, const char \*second);

\_Bool insert(const char \*src, const char \*str, int index, char \*dest, int len);



Рисунок 1 – Результат работы с программой (Задание 1)



Рисунок 2 – Результат работы с программой (Задание 2)



Рисунок 3 – Результат работы с программой (Задание 3)



Рисунок 4 – Результат работы с программой (Задание 4)



Рисунок 5 – Результат работы с программой (Задание 5)



Рисунок 6 – Результат работы с программой (Задание 6)