|  |
| --- |
| МИНОБРНАУКИ РОССИИ |
| Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА – Российский технологический университет»**  **РТУ МИРЭА** |
| Институт искусственного интеллекта |
| Кафедра технологий искусственного интеллекта |

ПРАКТИЧЕСКАЯ РАБОТА № 12

по дисциплине

«Процедурное программирование»

Обучающийся Сысоенко Глеб Максимович

Группа КАБО-01-23

Руководитель *Яковлев Д. А*

Москва 2023

**Практическая работа №12.**

**Тема: «Строки и символьные массивы в языке Си»**

**Цель лабораторной работы:**

Целью данной лабораторной работы освоить на практике работу со строками и символьными массивами.

**Описание:**

Составьте и выполните программу для операций со строками и символьными массивами в на языке программирования Си, согласно вариантам заданий пункта 7.

**Задание 1:**

Функция удаляет из строки слово с заданным номером.

**Код программы 1:**

//3.Функция удаляет из строки слово с заданным номером

#include <stdio.h>

#include <string.h>

#include <unistd.h>

#include <stdlib.h>

#include <time.h>

#include "math.h"

int stalin(char \* buffer){

int r, c = 0, xex = 0;

int \*A, \*B;

A = calloc(10, sizeof(int));

B = calloc(5, sizeof(int));

for(int i = 0; i < strlen(buffer); i++){

if (buffer[i] == ' '){

xex += 1;

if (buffer[i + 1] != ' ' && xex == 1){

A[c] = i;

c += 1;

}

else{

if (buffer[i + 1] != ' '){

A[c] = i;

A[c + 1] = i;

c += 2;

}

}

}

}

printf("\n\n");

int W1, W2, W3, W4, W5, i, cc;

for ( i = 0; i < 10; i++){

printf("| %2d | ", A[i]);

}

printf("\n");

B[0] = A[1] - A[0];

B[1] = A[3] - A[2];

B[2] = A[5] - A[4];

B[3] = A[7] - A[6];

B[4] = A[9] - A[8];

printf("\n");

cc = 1;

for (i = 0; i < 5; i++){

printf("DLINA DOSTUP SLOVA - > | %2d | NOMER SLOVA - > | %2d |\n", B[i], cc);

cc += 1;}

printf("\nVVEDITE NOMER SLOVA DLA YDOLENIA - > ");

scanf("%d", &r);

if (r == 1){

for(i = A[0]; i < A[1]; i++){

buffer[i] = ' ';}}

if (r == 2){

for(i = A[2]; i < A[3]; i++){

buffer[i] = ' ';}}

if (r == 3){

for(i = A[4]; i < A[5]; i++){

buffer[i] = ' ';}}

if (r == 4){

for(i = A[6]; i < A[7]; i++){

buffer[i] = ' ';}}

if (r == 5){

for(i = A[8]; i < A[9]; i++){

buffer[i] = ' ';}}

printf("\n%s\n", buffer);

}

int main(void)

{ // строка для записи

char \* message = " TEST MESSAGE ONLY TO TEST ";

// файл

char \* filename = "UMBRELA.inc.txt";

// запись в файл

FILE \*fp = fopen(filename, "w");

if(fp){

// записываем строку

fputs(message, fp);

fclose(fp);

printf("FILE ZAPISAN\n");}

// буфер для считавания данных из файла

char buffer[1024];

// чтение из файла

FILE \*p = fopen(filename, "r");

char \* string;

if(fp){

// пока не дойдем до конца, считываем по 256 байт

while((fgets(buffer, 1024, p))!=NULL)

{printf("%s", buffer);}

fclose(p);}

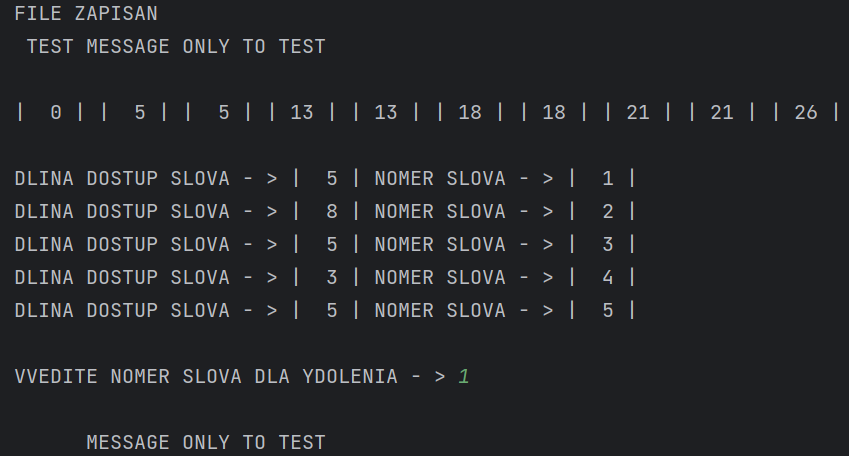
stalin(buffer);

if ((int)strlen(buffer) == 1024) // Clear buffer only when more symbols provided.

while (getchar() != '\n'); // For clearing stdin when we read less than provided.

}

**Скриншот выполнения программы 1:**



**Задание 2:**

Функция переводит строку, содержащую десятичное представление целого числа,

в строку, содержащую его шестнадцатиричное представление.

**Код программы 2:**

#include <stdio.h>

#include <stdlib.h>

//Функция переводит строку, содержащую десятичное представление целого числа,

//в строку, содержащую его шестнадцатиричное представление.

char\* RENOVATION\_STALIN(const char\* vx\_chislo) {

int num = atoi(vx\_chislo);

int dr\_ch;

int cel\_ch;

char hex[100] = {0};

int i = 0;

while (num != 0) {

dr\_ch = num % 16;

cel\_ch = num / 16;

if (dr\_ch < 10)

hex[i] = dr\_ch + '0';

else

hex[i] = dr\_ch + 55;

i++;

num = cel\_ch;}

char\* fn\_chislo = (char\*)malloc(sizeof(char) \* (i + 1));

int j = 0;

while (i > 0) {

i--;

fn\_chislo[j] = hex[i];

j++;}

fn\_chislo[j] = '\0';

return fn\_chislo;

}

int main() {

char vx\_chislo[100];

printf("VVEDITE CHISLO V 10 SS: ");

scanf("%s", vx\_chislo);

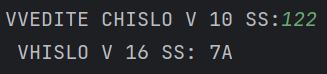
char\* fn1\_chislo = RENOVATION\_STALIN(vx\_chislo);

printf("VHISLO V 16 SS: %s\n", fn1\_chislo);

return 0;

}

**Скриншот выполнения программы 2:**



**Контрольные вопросы:**

1. Воспользоваться функциями malloc или сalloc.

2. Иначе программа будет работать не корректно и в конце концов это может привести к тому, что память на «машине» просто закончиться.

3. Malloc берёт и выделяет память, а calloc ещё и «чистит» занимаемый участок.

4. Очистка памяти после выполнения действий с динамическими массивами.

5. Изменение размерности массива по ходу выполнения программы.

6. Оптимизация затрачиваемых ресурсов на выполнение программы, когда надо «забираем» под свои нужды, когда надо освобождаем или изменяем требуемый нам размер.