МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное бюджетное образовательное учреждение

высшего образования

**«УЛЬЯНОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

ФАКУЛЬТЕТ ИНФОРМАЦИОННЫХ СИСТЕМ И ТЕХНОЛОГИЙ

Кафедра «Информационные системы»

Дисциплина «Основы алгоритмизации и программирования»

Лабораторная работа №11

Символы, Строки, Текстовые файлы

Выполнил:

студент гр. ПИбд-12

Дозорова Алена

Ульяновск, 2021 г

Задание:

1. **Hello по русски!**
2. Ввести символ CH. Вывести на экран 20 символов с кодами от CH до CH+19
3. Вывести на экран 30 символов с кодами от CH до CH-29
4. Ввести строку s. Подсчитать, сколько в ней пробелов
5. Ввести строку s. Все пробелы в ней заменить символом '#'
6. Ввести строку s. Все цифры в ней заменить символом ‘$‘.
7. Ввести строку s (без русских символов). Все маленькие латинские буквы превратить в большие
8. Создать на основе разработанного кода проект с несколькими модулями
9. Реализовать собственную версию функции int strlen(char \*)
10. Реализовать собственную версию функции int strcmp (char \*, char \*)
11. Реализовать собственную версию функции void strcpy (char \*, char \*)
12. Реализовать собственную версию функции void strcat (char \*, char \*)
13. Создать программно файл с русским текстом
14. Преобразовать файл in1.txt в файл out1.txt, заменив все маленькие буквы на большие
15. Преобразовать файл in1.txt в файл out2.txt, заменив все большие буквы на маленькие
16. Преобразовать текстовый файл in1.txt в файл out6.txt, в котором каждое слово выводится на отдельной строке
17. В файле in17.txt есть строки со словами в каждой строке ровно одно слово. С консоли вводится комплект букв. Нужно вывести в консоль те слова из файла in14.txt, которые можно составить из введенных с консоли букв
18. Все цифры в файле заменить символом ‘X’.

Код:

Laba11.cpp

// Laba11.cpp : Этот файл содержит функцию "main". Здесь начинается и заканчивается выполнение программы.

//

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <Windows.h>

#include "MyFunc.h"

void task1(char name[12])

{

printf("Приветствую тебя, %s\n", name);

}

void task2(char ch)

{

for (int c = ch; c <= ch + 19; c++)

{

printf("'%c' %d\n", c, c);

}

printf("\n\n\n");

}

void task3(char ch)

{

for (int c = ch; c >= ch - 29; c--)

{

printf("'%c' %d\n", c, c);

}

printf("\n\n\n");

}

void task4(char s[80])

{

int cnt = 0;

for (int i = 0; i < strlen(s); i++)

{

if (s[i] == ' ') cnt++;

}

printf("\nВ ней %d пробелов\n", cnt);

}

void task5(char s[80])

{

for (int i = 0; i < strlen(s); i++)

{

if (s[i] == ' ') s[i] = '#';

}

printf("Вы получили строку s = \"%s\"", s);

}

void task6(char s[80])

{

printf("Вы ввели строку s = \"%s\"", s);

for (int i = 0; s[i] != '\0'; i++)

{

if (isdigit(s[i])) s[i] = '$';

}

printf("Вы получили строку s = \"%s\"", s);

}

void task6\_2(char s[80])

{

printf("Вы ввели строку s = \"%s\"", s);

for (int i = 0; s[i] != '\0'; i++)

{

if (IsDigit(s[i])) s[i] = '$';

}

printf("Вы получили строку s = \"%s\"", s);

}

void task7(char s[100])

{

for (int i = 0; s[i] != '\0'; i++)

{

s[i] = toupper(s[i]);

}

printf("Вы получили строку s = \"%s\"", s);

}

void task7\_1(char s[100])

{

for (int i = 0; s[i] != '\0'; i++)

{

s[i] = ToUpper(s[i]);

}

printf("Вы получили строку s = \"%s\"", s);

}

void task10(char\* s, char\* s1)

{

printf("%d\n", StrCmp(s, s1));

if (StrCmp(s, s1) == strcmp(s, s1))

{

printf("Работает верно");

}

else

{

printf("Работает !верно");

}

}

int main()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

char s[80];

char s1[100];

printf("Введите строку/символ: ");

fgets(s, 79, stdin);

printf("Вы ввели s = %s", s);

printf("Введите строку 2: ");

fgets(s1, 99, stdin);

printf("Вы ввели s1 = %s", s1);

int x;

x = 0;

while (x != -1)

{

printf("\n1: task1\n2: task2\n3: task3\n4: task4\n5: task5\n6: task6\n6.1: task6.1\n7: task7\n7.1: task7.1\n9: task9\n11: task11\n12: task12\n");

scanf("\n%d", &x);

switch (x)

{

case 1:

task1(s);

break;

case 2:

task2(s[0]);

break;

case 3:

task3(s[0]);

break;

case 4:

task4(s);

break;

case 5:

task5(s);

break;

case 6:

task6(s);

break;

case 62:

task6\_2(s);

break;

case 7:

task7(s);

break;

case 72:

task7\_1(s);

break;

case 9:

printf("%d", StrLen(s));

break;

case 10:

task10(s, s1);

break;

case 11:

printf("%s\n", StrCpy(s1, s));

if (StrCpy(s1, s) == strcpy(s1, s))

{

printf("Работает верно");

}

else

{

printf("Работает !верно");

}

break;

case 12:

printf("%s\n", StrCat(s1, s));

if (StrCat(s1, s) == strcat(s1, s))

{

printf("Работает верно");

}

else

{

printf("Работает !верно");

}

break;

}

}

{

int x;

scanf("%d", &x);

}

}

MyFuck.cpp

#pragma once

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <stdio.h>

#include <Windows.h>

#include "MyFunc.h"

bool IsDigit(char c)

{

if (c >= '0' && c <= '9')

{

return true;

}

return false;

}

char ToUpper(char c)

{

char new\_c;

if (c >= 'a' && c <= 'z')

{

new\_c = 'A' + (c - 'a');

}

if (c >= 'а' && c <= 'п')

{

new\_c = 'А' + (c - 'а');

}

if (c >= 'р' && c <= 'я')

{

new\_c = 'Р' + (c - 'р');

}

if (c == 'ё')

{

new\_c = 'Ё';

}

return new\_c;

}

int StrLen(char s[1000])

{

int n = 0;

for (int i = 0; s[i] != '\0'; i++)

{

n++;

}

return n-1;

}

int StrCmp(char s1[100], char s2[100])

{

if (StrLen(s1) > StrLen(s2))

{

return 1;

}

else if (StrLen(s1) > StrLen(s2))

{

return -1;

}

else

{

for (int i = 0; s1[i] != '\0'; i++)

{

if (s1[i] != s2[i])

{

if (s1[i] > s2[i])

{

return 1;

}

else

{

return -1;

}

}

}

return 0;

}

}

char\* StrCpy(char\*s\_copy, char\* s)

{

for (int i = 0; s[i] != '\0'; i++)

{

s\_copy[i] = s[i];

}

return s\_copy;

}

char\* StrCat(char\* s, char\* s1)

{

int i = 0;

while(s1[i] != '\0')

{

i++;

}

for (int j = i; s1[j] != '\0'; j++)

{

s[i] = s1[j];

}

return s;

}