Министерство образования Республики Беларусь

Учреждение образования

«Брестский государственный технический университет»

Кафедра ИИТ

Лабораторная работа №3

за 4 семестр

По дисциплине: «КСиС»

Тема: «Язык ассемблера. Обработка символьных данных»

Выполнил:

Студент 2 курса

Группы ПО-4(1)

Иваненко И. Л.

Проверил:

Савицкий Ю. В.

2021

Лабораторная работа №3

Язык ассемблера. Обработка символьных данных

Цель работы: требуется написать на ассемблере программу обработки текста.

Вариант 10

**Задание:**

Проверяемое условие: Количество строчных латинских букв равно сумме цифр в нем.

Первое правило преобразования: Заменить каждую ненулевую цифру соответствующей ей строчной буквой латинского алфавита(1 –> a, 2-> b)

Второе правило преобразования: Удалить элемент на каждой второй позиции

**Код программы:**

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <iostream>

using namespace std;

int getLowerCaseCharsCount(char\* str) {

int lowCount = 0;

\_asm {

mov esi, str

\_loop:

mov al, [esi]

or al, al

jz \_ext

cmp al, 123

jb \_below123

jmp \_nxt

\_below123:

cmp al, 97

jae \_lowercase

\_nxt:

inc esi

jmp \_loop

\_lowercase:

inc lowCount

jmp \_nxt

\_ext:

}

return lowCount;

}

int getDigitsSum(char\* str) {

int digSum = 0;

\_asm {

mov esi, str

\_loop:

mov al, [esi]

or al, al

jz \_ext

cmp al, 58

jb \_below58

jmp \_nxt

\_below58:

cmp al, 48

jae \_digit

\_nxt:

inc esi

jmp \_loop

\_digit:

movzx eax, al

sub eax, '0'

add digSum, eax

jmp \_nxt

\_ext:

}

return digSum;

}

void changeChars(char\* str) {

\_asm {

mov esi, str

\_loop:

mov al, [esi]

or al, al

jz \_ext

cmp al, 58

jb \_below58

jmp \_nxt

\_below58:

cmp al, 48

ja \_digit

\_nxt:

inc esi

jmp \_loop

\_digit:

add al, 48

mov [esi], al

jmp \_nxt

\_ext:

}

}

char\* delEachSecPosition(char\* str) {

int len = strlen(str);

\_asm {

mov esi, str

mov edi, esi

mov ecx, len

shr ecx, 1

mov ebx, ecx

\_loop:

lodsw

stosb

loop \_loop

mov ecx, ebx

xor al, al

rep stosb

}

}

int main() {

setlocale(0, "");

const size\_t MAX\_STRING\_LENGTH = 100;

char\* str1 = new char[MAX\_STRING\_LENGTH];

strcpy(str1, "abA\_cdDe[]WfghJi13j8kl");

if (str1 != "") {

cout << " Строка 1: " << str1 << endl;

cout << " Количество строчных латинских букв: " << getLowerCaseCharsCount(str1) << endl;

cout << " Сумма цифр в строке: " << getDigitsSum(str1) << endl;

changeChars(str1);

cout << " Выполняем первое правило преобразования: " << str1 << endl;

}

cout << endl;

char\* str2 = new char[MAX\_STRING\_LENGTH];

strcpy(str2, "ajdiwjhdiowhf34234HAFWHFUF");

if (str1 != "") {

cout << " Строка 2: " << str2 << endl;

cout << " Количество строчных латинских букв: " << getLowerCaseCharsCount(str2) << endl;

cout << " Сумма цифр в строке: " << getDigitsSum(str2) << endl;

delEachSecPosition(str2);

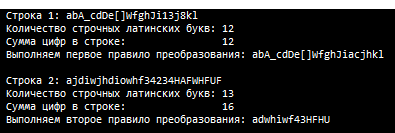
cout << " Выполняем второе правило преобразования: " << str2 << endl;

}

return 0;

}

**Результат выполнения:**



Вывод: В ходе выполнения данной лабораторной работы я научился обрабатывать текст на языке Ассемблера.