

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА(ЛЕНИНА)
КАФЕДРА МО ЭВМ

ОТЧЕТ

по лабораторной работе №4

по дисциплине «Организация ЭВМ и систем»

Тема: Представление и обработка целых чисел. Организация ветвящихся процессов

Студентка гр. 1381

Деркачева Д.Я.

Преподаватель

Ефремов М. А.

Санкт-Петербург

2022

Цель работы

Представить и обработать символьную информацию с использованием строковых команд на языке ассемблера.

Текст задания

Разработать программу обработки символьной информации, реализующую функции:

- инициализация (вывод титульной таблички с указанием вида преобразования и автора программы) - на ЯВУ;
- ввода строки символов, длиной не более N_{\max} (≤ 80), с клавиатуры в заданную область памяти - на ЯВУ; если длина строки превышает N_{\max} , остальные символы следует игнорировать;
- выполнение заданного в таблице 5 преобразования исходной строки с записью результата в выходную строку - на Ассемблере;
- вывода результирующей строки символов на экран и ее запись в файл - на ЯВУ.

Ассемблерную часть программы включить в программу на ЯВУ по принципу встраивания (in-line).

Ход выполнения работы

Вариант 7 - Инвертирование введенных во входной строке цифр в восьмеричной СС и преобразование заглавных русских букв в строчные, остальные символы входной строки передаются в выходную строку непосредственно.

При выполнении работы сначала были инициализированы два массива, в одно будут лежать изначальные значения, во втором измененные в ходе модификации

символов по заданию. В *main()* сначала выводится информация о выполняющем работу и задание, после происходит считывание с помощью *fgets*, дальше был прописан ассемблерный модуль в котором происходит проверка считываемых символов по метке *check*:. Здесь символы проверяются на верхнестрочный регистр, если такое значение обнаружилось, то оно изменяется на нижний регистр, если нет, дальше следует блок с меткой *not_uppercase*, здесь проверяется введенный символ на соответствие с “ё”, т.к. данной буквы нет в аски таблице, если это и не этот символ то программа переходит к метке *not_yo*:. Здесь происходит инвертирование цифр меньше 8, т.к. по заданию мы работаем с цифрами из *8cc*, если были введены другие цифры, то они просто записываются, собственно также происходит и с иными символами не являющимися русскими буквами. Последняя метка в данном модуле *final*:, которая отправляет программу обрабатывать последующие введенные символы, если они не были концом строки. Завершающим шагом в программе является вывод измененного сообщения в консоль и сохранение его в файле.

Текст исходного файла программы lab4.cpp

Текст исходной программы lab4.cpp см. в приложении А..

Тестирование

Табл.1. Результаты тестирования

№ п/п	Входные данные	Выходные данные	Комментарий
1	ПриВЕт АндрЕ71	привет андре06	Успешное завершение программы
2	Холодный 520 вечччЕр	холодный 257 вечччер	Успешное завершение программы
3	Хах3241зАААЦЁё	хах4536зааацёё	Успешное завершение

			программы
--	--	--	-----------

Выводы по работе

В ходе выполнения лабораторной работы была написана программа на c++ с ассемблерной вставкой, которая преобразует символы в соответствии с заданием.

Приложение А

Текст исходного файла lab4

```
#include <iostream>
#include <stdio.h>
#include <windows.h>

char in_str[81];
char out_str[81];

int main()
{
    std::cout << "Derkacheva Darya from 1381 group\n";
    std::cout << "task 7: *convert uppercase to lowercase";

    std::cout << " *invert numbers in 8 number system" <<
std::endl;

    SetConsoleCP(1251); //input rus symbol
    SetConsoleOutputCP(1251);
    fgets(in_str, 81, stdin);
```

```

in_str[strlen(in_str) - 1] = '\\0';

__asm {
    push ds
    pop es
    mov esi, offset in_str
    mov edi, offset out_str
    check:
        lodsb //use esi
        cmp al, 192 //big first
        jnl not_uppercase
        cmp al, 223 //big last
        jg not_uppercase
        add al, 32 //go to lowercase
        stosb
        jmp final

    not_uppercase:
        cmp al, 'Ë' //not in ascii
        jne not_yo
        mov al, 'ë'

    not_yo:
        cmp al, 48
        jnl not_between_zero_and_seven
        cmp al, 55
        jg not_between_zero_and_seven
        neg al

```

```

        add al, 103

        stosb

        jmp final

not_between_zero_and_seven:

        stosb

final:

        mov ecx, '\0'

        cmp ecx, [esi]

        je  checkExit

        jmp check

checkExit:

};

std::cout << out_str;

FILE* f;

fopen_s(&f, "out.txt", "w");

fwrite(out_str, sizeof(char), strlen(out_str), f);

fclose(f);

return 0;

}

```