

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 символьной информации с
использованием строковых команд.

Студентка гр. 1303

Хабибуллина А.М.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2022

Цель работы.

Разработать на языке Ассемблера программу обработки символьной информации.

Задание.

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).

Вариант 25:

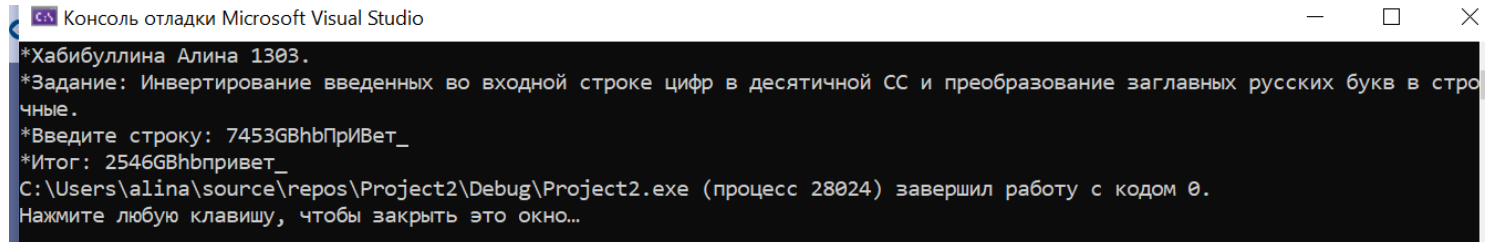
Инвертирование введенных во входной строке цифр в десятичной СС и преобразование заглавных русских букв в строчные, остальные символы входной строки передаются в выходную строку непосредственно.

Выполнение работы

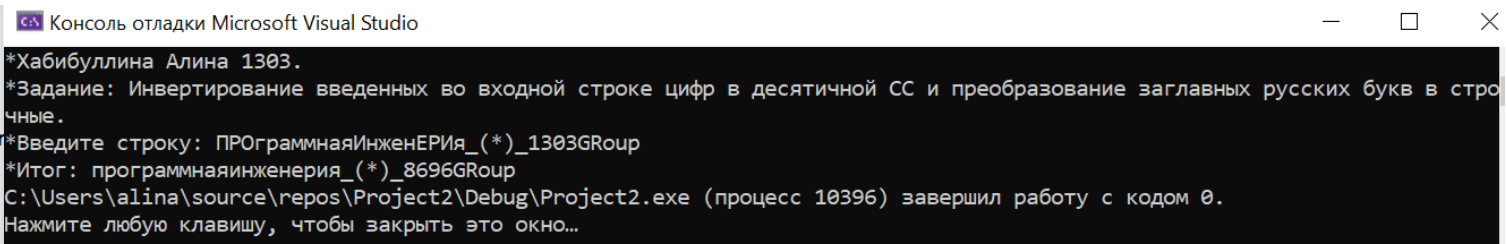
Создаем два массива для входной и выходной строки. Производится ввод строки и открытие файла, куда будет записываться итоговая строка с преобразованиями. В блоке `__asm` производим следующие операции для преобразования строки. В регистры `esi`, `edi` присваиваются переменные `input_str` и `output_str`. Команда `lodsb` осуществляет посимвольное считывание строки. Далее происходит проверка символа: если это конец строки, то блок завершается, если это десятичная цифра, то происходит переход к метке `inverse`, где данная цифра инвертируется, если символ – заглавная русская

буква, то осуществляется переход к метке change, где заглавная буква меняется на строчную. С помощью команды stosb символ записывается в регистр edi. После завершения блока, полученная строка с преобразованиями выводится в консоль и в файл.

Тестирование



```
Консоль отладки Microsoft Visual Studio
*Хабибуллина Алина 1303.
*Задание: Инвертирование введенных во входной строке цифр в десятичной СС и преобразование заглавных русских букв в строчные.
*Введите строку: 7453GBhbПРИвет_
*Итог: 2546GBhbпривет_
C:\Users\alina\source\repos\Project2\Debug\Project2.exe (процесс 28024) завершил работу с кодом 0.
Нажмите любую клавишу, чтобы закрыть это окно...
```



```
Консоль отладки Microsoft Visual Studio
*Хабибуллина Алина 1303.
*Задание: Инвертирование введенных во входной строке цифр в десятичной СС и преобразование заглавных русских букв в строчные.
*Введите строку: ПРОграммнаяИНженЕРия_(*)_1303GRoup
*Итог: программнаяинженерия_(*)_8696GRoup
C:\Users\alina\source\repos\Project2\Debug\Project2.exe (процесс 10396) завершил работу с кодом 0.
Нажмите любую клавишу, чтобы закрыть это окно...
```

Выводы

В ходе выполнения лабораторной работы были получены навыки разработки программы, выполняющей обработку символьной информации с использованием строковых команд, на языке программирования Ассемблер.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: main.cpp

```
#include <iostream>
#include <fstream>
#include <windows.h>
using namespace std;
char input_str[81];
char output_str[81];

int main() {

    SetConsoleCP(1251);
    SetConsoleOutputCP(1251);

    cout << "*Хабибуллина Алина 1303.\n*Задание: Инвертирование
введенных во входной строке цифр в десятичной СС и
преобразование заглавных русских букв в строчные.\n";
    cout << "*Введите строку: ";
    cin.getline(input_str, 81);

    ofstream file;
    file.open("out.txt");

    __asm {
        push ds
        pop es
        mov esi, offset input_str
        mov edi, offset output_str

        check :
            lodsb
            cmp al, '\0'
            je finish

            cmp al, '0'
            jb symbol

            cmp al, '9'
            jbe inverse

            cmp al, 'Ё'
            je sym

            cmp al, 'А'
            jb symbol

            cmp al, 'Я'
            jbe change
```

```

        cmp al, 'Я'
        jg symbol

sym :
    add al, 16
    jmp symbol

change :
    add al, 32
    jmp symbol

inverse :
    neg al
    add al, 105

symbol :
    stosb
    jmp check

finish :

};

cout << "ИТОГ: " << output_str;
file << output_str;
file.close();

return 0;
}

```