

**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. 0382

\_\_\_\_\_

Деткова А.С

Преподаватель

\_\_\_\_\_

Ефремов М.А.

Санкт-Петербург

2021

### **Цель работы.**

Научиться обрабатывать строки и символы на языке Ассемблера.

### **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  $N_{\max}$  ( $\leq 80$ ), 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 3:

Формирование выходной строки только из русских и латинских букв входной строки

### **Выполнение работы.**

Программа написана на языке C++, в которой включена ассемблерная часть. В начале работы задается локаль для работы с русским языком, выводится информация об авторе программы и задание. Работаем с двумя массивами: *inp* – массива входной строки (длиной 80) и *res* – массива выходной строки (длина также 80). Строка длиной не больше 80 символов считывается с помощью функции *getline*.

Ассемблерная часть.

*edi* – индексный регистр назначения, с помощью него происходит индексация по выходному массиву данных. *esi* – индексный регистр источника, в нем содержится адрес источника. В начале делается проверка символа, если встречен символ конца строки '\0', то обработка строки прекращается. Сравниваются ASCII-коды символов, английские заглавные лежат в диапазоне [65,90], английские строчные — [97,122], русские — [192,255]. Если обрабатываемый символ попадает в этот диапазон, то он записывается в выходную строку, если нет, то переходим к след. символу входной строки. Все переходы осуществляются с помощью условных и безусловных переходов по меткам. Если символ  $\geq 192$ , переход на метку *firstCondForRus* (русские буквы), если символ  $\geq 97$  – *firstCondForEngTwo* (английские строчные), если символ  $\geq 65$  – *firstCondForEngOne* (английские заглавные). На метке *firstCondForRus*: если символ  $\leq 255$ , переход на метку *secondCond*. На метке *firstCondForEngTwo*: если символ  $\leq 122$ , переход на метку *secondCond*. На метке *firstCondForEngOne*: если символ  $\leq 122$ , переход на метку *secondCond*. На метке *secondCond*: записываем символ в строку выхода, увеличиваем индексный регистр. Переход на метку *result*: переход к следующему символу входной строки, переход в начало на метку *start*. Если встречен '\0' – конец строки, то переход на метку *end* – конец ассемблерной части программы.

В конце программы вывод строки-результата.

Разработанный программный код см. в приложении А.

## Тестирование.

1. xkjcnвырпвп729873497dhfjk././,+ -//.ов

Результат:

хкјснвырпвпdhfјков

2. 00008883щщщhjs hd+\*#589&^^^/^\jsjdn

Результат:

ooooщщщhjsjdjn

3. hhhhhhhhhh888888888888888888reeeeeeeeemmmmmmmmmmm

dlllllllllllpaodoskfv;lksv

Результат:

# hhhhhhhhhhreeeeeeeeeemmmmmmmmmmdllllllllllpaodoskfvksv

4. jddddd75555555555555555555

[illegible]

222222222222222vnpmooooooooooooooooo[[]]]]]]]]]]]]]]]][[]][[[[{ } { } }

{}xkjdifuu984348fvhkjnss,s,

Результат:

jddddddddddddddddddddddnnvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv

Лишние символы отброшены.

## Выводы.

Были изучены способы обработки строк и символов на языке Ассемблера. Была разработана программа, которая из строки длиной не более 80 символов составляет новую строку, в которой остаются только буквы русского и английского алфавита.

## ПРИЛОЖЕНИЕ А

### ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: lab4.cpp

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

int main() {
    SetConsoleCP(1251);
    SetConsoleOutputCP(1251);
    std::cout << "Деткова Анна, группа 0382" << std::endl <<
        "Вариант 3: Формирование выходной строки только из
русских и латинских букв входной строки" << std::endl;
    char inp[80];
    char res[80] = {80 * '\0'};
    std::cin.getline(inp, 80);

    __asm {
        mov edi, 0
        lea esi, inp // загружаем адрес начала массива в
индексный регистр источника - esi

        start:
        mov ah, [esi] // массив букв - байтовый регистр,
загружаем в верхнюю
        // половину базового регистра содержимое по
адресу

        cmp ah, 0 // если конец строки
        je end

        cmp ah, 192
        jae firstCondForRus

        cmp ah, 97
        jae firstCondForEngTwo

        cmp ah, 65
        jae firstCondForEngOne

        jmp result

        firstCondForRus :
        cmp ah, 255
        jbe secondCond
        jmp result

        firstCondForEngTwo:
        cmp ah, 122
        jbe SecondCond
        jmp result

        firstCondForEngOne :
        cmp ah, 90
        jbe SecondCond
```

```

        jmp result

secondCond :
mov res[edi], ah
inc edi

result :
inc esi
jmp start
end :
}

std::cout << std::endl << std::endl;
std::cout << "Результат:" << std::endl;
puts(res);
std::cout << std::endl << std::endl;

return 0;
}

```