

**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. 9382

\_\_\_\_\_

Пя С.

Преподаватель

\_\_\_\_\_

Ефремов М.А.

Санкт-Петербург

2020

## **Цель работы.**

Изучить команды для работы со строками ассемблера, написать программу, обрабатывающую вводимую строку определенным способом и познакомиться с принципом встраивания in-line на примере ЯВУ C++.

## **Задание:**

### **15 Вариант**

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).

15. Исключение русских букв и цифр, введенных во входной строке, при формировании выходной строки.

## **Ход работы:**

При разработке программы были использованы следующие команды:

LODSB - копирует один байт из памяти по адресу DS:SI в регистр AL. После выполнения команды, регистр SI увеличивается на 1, если флаг DF = 0, или уменьшается на 1, если DF = 1.

STOSB - сохраняет регистр AL в ячейке памяти по адресу ES:DI. После выполнения команды, регистр DI увеличивается на 1, если флаг DF = 0, или уменьшается на 1, если DF = 1.

CLD - очищает флаг направления (DF). Такая необходимость может возникнуть при работе с цепочечными командами.

### Тестирование.

Вводные данные	Результат
42укцкцЁЁЁЁЁЁЁЁёйукцк кцк!цавкцуккоукуцквукуцк54535е 4242уыфваыуоукцк4234уу!	i love you!
1231313132	
123qeqweqwe(при n = 5)	qe

### Выводы.

В результате выполнения лабораторной работы был разработан код для определенной обработки строк. Были улучшены навыки письма в ассемблере.

### Приложение.

#### Текст файла main.cpp

```
#include <iostream>
#include <fstream>

#define n 80
int main() {
    system("chcp 1251 > nul");
    setlocale(LC_STYPE, "rus");
    std::cout << "Вариант 15.\nИсключение русских букв и цифр, введенных во\nвходной строке, при формировании выходной строки.\nстудент 9382 г. - Пя Сон\nХва\n";
    char str[n + 1];
    char answer[n + 1];
    std::cout << "Введите строку для обработки:\n";
    std::cin.getline(str, n + 1);
```

```

std::cout << "Строка до обработки:\n" << str << "\n";
bool flag = false;
for (int i = 0; i < strlen(str); i++)
    if (isalpha(str[i]))
        flag = true;
    if (flag) {
        _asm{
            mov ecx, n; длина строки в ecx
            mov al, 0
            lea si, str; кладем в ds:si адрес str
            lea di, answer; кладем в di адрес answer
            cld; обнуление флага направления

            data_processing:
            lodsb; копирует один байт из памяти по адресу ds:si в
регистр al

            cmp al, 'ё'
            je for_exception; исключение кириллицы
            cmp al, 'Ё'
            je for_exception
            cmp al, 'А'
            jl check_digit_case
            cmp al, 'Я'
            jg check_digit_case
            loop data_processing
            jmp finish_processing

            for_exception:
            loop data_processing

            check_digit_case::; исключение цифр
            cmp al, '0'
            jl add_to_answer
            cmp al, '9'
            jg add_to_answer
            loop data_processing

            add_to_answer:
            stosb; сохраняет регистр al в ячейке памяти по адресу
es:di

            loop data_processing

            finish_processing:
            mov al, 0
            stosb
        }
    } else {
        answer[0] = '\0';
    }
std::cout << "Вывод обработанной строки:\n" << answer;
std::fstream fout("output.txt");
fout << "Строка до обработки:\n" << str << "\nВывод обработанной
строки:\n" << answer;
return 0;
}

```