

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

**ОТЧЕТ**  
**по лабораторной работе №4**  
**по дисциплине «Организация систем и ЭВМ»**  
**Тема «Представление и обработка символьной информации с**  
**использованием строковых команд.»**

Студент гр. 1383

Валиев Р.Р.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2022

### **Цель работы.**

Разработать на языке Ассемблера программу, обрабатывающую символьную информацию с использованием строковых команд.

### **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 2:**

Формирование выходной строки только из цифр и латинских букв входной строки.

### **Выполнение работы.**

Выполнение лабораторной работы проводилось на языке C++.

В начале на ЯВУ выводится имя, номер группы и само задание. Далее считывается 81 символьный знак или меньше из потока ввода (включая терминирующий ноль) с помощью команды *fgets*.

В ассемблерном блоке, объявляемым ключевым словом *\_asm*, происходит посимвольное считывание введённой строки с помощью команды *lodsb*, помещающая считавшийся из *es:esi* байт в *al*. Далее несколькими

командами *str* проверяется вхождение символа в промежуток '0-9', 'A'-'Z' и 'a'-'z'. При выходе за эти диапазоны символ игнорируется и читается следующий, иначе вызывается команда *stosb*, которая записывает байт *al* в *es:edi* – где изначально хранится отступ выходной строки. Ассемблерный код заканчивается, если встречается терминирующий нуль.

После, на ЯВУ, полученная строка выводится на экран и в текстовый файл с помощью функций стандартной библиотеки языка C++.

Разработанный программный код см. в приложении А.

Таблица 1 – Результаты тестирования

Входные данные	Результат
123 !@# qwe QWE ёйцу ЁЙЦУ	123qweQWE
Абв Abc 123 YUvvs	Abc123YUvvs
!345 wjh UU абвгд	345wjhUU

### **Вывод.**

В результате лабораторной работы была изучена обработка символьной информации с использованием языка ассемблера, а также разработана программа на ЯВУ, использующая вставку на языке ассемблера.

## ПРИЛОЖЕНИЕ А

### ИСОДНЫЙ КОД ПРОГРАММЫ.

Название файла: **lab4.cpp**

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

```
char input_string[81];
char output_string[81];
```

```
int main() {
    setlocale(LC_CTYPE, "rus");
    std::cout << "Валиев Р.Р.\nВариант 2. Формирование выходной строки
только из цифр и латинских букв входной строки\n";
```

```
    fgets(input_string, 81, stdin);
    input_string[strlen(input_string)] = '\0';
```

```
    __asm {
        push ds
        pop es
        mov esi, offset input_string
        mov edi, offset output_string
        read :
        lodsb
```

```
        cmp al, '0'
        jl next
        cmp al, '9'
        jle write
```

```
        cmp al, 'A'
        jl next
        cmp al, 'Z'
```

```
jle write
```

```
    cmp al, 'a'  
    jl next  
    cmp al, 'z'  
    jg next
```

```
    write :  
    stosb  
    next :  
    cmp[esi], '\0'  
    jne read  
};
```

```
std::cout << output_string << std::endl;  
std::ofstream file("output.txt");  
file << output_string;  
std::cin;  
return 0;  
}
```