

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

Студент гр. 0383

Преподаватель

Сергеев Д.В.

Ефремов М.А.

Санкт-Петербург

2021

Цель работы.

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 14:

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

Выполнение работы.

Программа реализована на языке C++ со вставкой на языке ассемблера.

В функции *main()* выводится титульная табличка с указанием вида преобразования и автора программы. Затем считывается строка, которая обрабатывается на языке ассемблера. При помощи команды *lodsb* загрузится символ в регистр AL. Если же символ - цифра или буква латинского алфавита, он пропускается. Если символ удовлетворяет условиям нашей программы, то он записывается в выходную строку при помощи команды *stosb*. Новая обработанная строка выводится в консоль и записывается в файл *output.txt*.

Результаты тестирования представлены в табл. 1.

Текст исходного файла программы см. в приложении А.

Таблица 1. Проверка работы программы.

№	Входная строка	Выходная строка	Комментарий
1	059 АВ ab ZY zy АЮ аб ЯЮ яю + -*///\	АЮ аб ЯЮ яю + -*///\	Результат верен

Выводы.

В ходе выполнения данной лабораторной работы была изучена работа со строками на языке ассемблера, а также использована возможность “inline” встраивания asm кода в C++.

ПРИЛОЖЕНИЕ А

ТЕКСТЫ ИСХОДНЫХ ФАЙЛОВ ПРОГРАММ

Название файла: **lab4.cpp**

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

char instr[81];
char outstr[81];

//059 AB ab ZY zy АЮ аб ЯЮ яю + -*///\\

int main() {

    fgets(instr, 81, stdin);

    //14) Исключение латинских букв и цифр, введенных во входной строке
    при формировании выходной строки.
    __asm {
        push ds
        pop es
        mov esi, offset instr
        mov edi, offset outstr
        cycle :
        lodsb; загрузить символ в AL
            // не устраивает 48 - 57, 65 - 90, 97 - 122

        cmp al, 48 ; 0
        jge step ; прыжок если первый >=
        stosb
        jmp check

        step1 :
        cmp al, 57 ; 9
        jbe check ; прыжок если первый <=
        cmp al, 65 ; A
```

```
jge step2
```

```
stosb
```

```
jmp check
```

```
step2 :
```

```
cmp al, 90 ; z
```

```
jbe check
```

```
cmp al, 97 ; a
```

```
jge step3
```

```
stosb
```

```
jmp check
```

```
step3 :
```

```
cmp al, 122 ; z
```

```
jbe check
```

```
stosb
```

```
jmp check
```

```
check :
```

```
mov ecx, '\0'
```

```
cmp ecx, [esi]
```

```
je lexit ; выход из цикла, если текущий символ завершающий
```

```
jmp cycle
```

```
lexit :
```

```
}
```

```
std::cout << outstr;
```

```
FILE* fout;
```

```
fopen_s(&fout, "output.txt", "w");
```

```
fputs(outstr, fout);
```

```
return 0;
```

```
}
```

