

**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. 0382

Злобин А.С.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2021

### **Цель работы.**

Научится обработке символьной информации с использованием строковых команд. Также научится встраивать ассемблерный код в код языков высокого уровня по принципу *in-line*.

### **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 1

Инвертирование введенных во входной строке цифр в десятичной системе счисления (СС) и преобразование строчных русских букв в заглавные, остальные символы входной строки передаются в выходную строку непосредственно.

### Замечания:

- 1) При выполнении преобразования обязательно использовать команды работы со строками;
- 2) При выполнении преобразования нельзя портить входную строку. Результат преобразования должен записываться в выходную строку.

### **Выполнение работы.**

Выделяем память под вводимую и выводимую строку:

```
char s[81];  
char outstr[161];
```

Функция `main()` выводит приветственное сообщение с автором программы и описанием преобразования которое она делает. Считывает вводимую строку , открывает файл для чтения :

```
std::cout << "Zlobin Andrew, group 0382, task: invert numbers" <<  
std::endl;  
SetConsoleCP(1251);  
SetConsoleOutputCP(1251);  
fgets(s, 81, stdin);
```

Затем идет код на ассемблере встроенный в программу по принципу *in-line* , которые преобразует входную строку в соответствии с заданием .

### Ассемблерная часть программы:

1. Сохраняем в индексные регистры смещение в базе данных для входной и выходной строки.

```
mov esi, offset s  
  
mov edi, offset outstr
```

2. Помечаем начало блока обработки текущего символа *check\_symb1* и используем команду *lodsб* , которая загружает в регистр AL байт из ячейки памяти, указываемой при помощи индексного регистра-источника.

L :

*lodsб*

3. Далее с помощью команды сравнения *сmp* и команд условного перехода определяется в каком диапазоне находится текущий символ в кодировке WIN-1251 (текущий символ хранится в AL). Если этот символ находится в диапазоне букв, переход осуществляется на метку сохранения символа в выходную строку. Аналогично с русскими буквами + ё и Ё. Если текущий символ "\0" программа завершает обработку и выходит из ассемблеровской части.

4. Блок ответственный за сохранение в выходную строку нужных символов. С помощью команды *stosб* , которая сохраняет регистр AL в ячейке памяти по адресу DS:EDI. И перенаправляет выполнения программы на обработку следующего символа.

5. Блок кода выхода из цикла обработки строки , выполняется когда встречен символ "\0". Записывает этот символ в выходную строку с помощью команды *stosб*.

После ассемблерной части обработки входной строки , программа выводит выходную строку в консоль и файл и закрывает файл.

### Тестирование.

Таблица 1 – Результаты тестирования

№ п/п	Входные данные	Выходные данные	Комментарии
1.	тест*123ТЕСТ	ТЕСТ*876ТЕСТ	Программа работает корректно
2.	123456789	876543210	Программа работает корректно

## Выводы.

Была изучена обработка символьной информации на языке ассемблер с использованием строковых команд.

В ходе данной лабораторной работы была разработана программа , которая формирует выходную строку только из цифр и русских букв входной строки. Программа написана на языке высокого уровня (C++), с использованием встроенного по принципу *in-line* ассемблерного кода .

## Приложение А

### Исходный код программы

Название файла: lb4.cpp

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

char s[81];
char outstr[161];

int main()
{
    std::cout << "Zlobin Andrew, group 0382, task: invert numbers" <<
std::endl;
    SetConsoleCP(1251);
    SetConsoleOutputCP(1251);
    fgets(s, 81, stdin);
    s[strlen(s) - 1] = '\\0';
    __asm {
        push ds
        pop es
        mov esi, offset s
        mov edi, offset outstr
        L :
        lodsb // в al очередной символ

                cmp al, 224
                jl not_uppercase
                cmp al, 255
                jg not_uppercase
                // Чтобы конвертировать русские заглавные буквы в
прописные,
                // необходимо сместиться на 32 бита вперед
```

```

        sub al, 32
        stosb
        jmp final

not_uppercase:

```

```

cmp al, 'ë'
jne not_yo
mov al, 'Ë'

```

```

not_yo :

```

```

cmp al, 48
jl not_between_zero_and_seven
cmp al, 57
jg not_between_zero_and_seven
neg al
add al, 105
stosb
jmp final

```

```

not_between_zero_and_seven:

```

```

stosb // кладем в выходную строку байт из al

```

```

        final:
mov     ecx, '\0'
        cmp     ecx, [esi]
        je      LExit // выход из цикла, если текущий символ
завершающий
        jmp     L
LExit :

```

```

};
std::cout << outstr;
FILE* f;
fopen_s(&f, "out.txt", "w");
fwrite(outstr, sizeof(char), strlen(outstr), f);
return 0;
}

```