

**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

Морева Е.С.

Преподаватель

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  $N_{\max}$  ( $\leq 80$ ), 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).

## **Задание.**

Вариант 9.

Реализовать преобразование введенных во входной строке десятичных цифр в восьмеричную СС, остальные символы входной строки передаются в выходную строку непосредственно.

## **Выполнение работы.**

Написание программы производилось на языках C++ и Ассемблер с использованием компилятора Microsoft Visual Studio.

Определяются глобальные переменные – два массива символов (strIn для считывания строки и strOut для записи результата). Для первого выделено памяти на 81 символ (по условиям входные данные до 80ти символов, и ещё один

нужен для символа завершения строки ‘\0’), для второго – 161 (это максимальное количество символов после преобразования, т.к. любая десятичная цифра займёт в переводе в восьмеричную систему не более двух символов). В главной функции *main* в первый массив из стандартного потока ввода *stdin* с помощью *fgets* считывается строка.

Далее начинается ассемблерная часть программы.

Которая обрабатывает введенную строку и, если находит в ней 8 или 9 заменяет на 10 и 11 соответственно, так как остальные цифры десятичной и восьмеричной системах счисления имеют одинаковое представление.

### **Тестирование.**

Введено: h76guhbj0098yrvy

Результат: h76guhbj01110yrvy

Введено: mc8348x0s84yi4o0

Результат: mc10348x0s104yi4o0

Введено: b876754b98

Результат: b1076754b1110

Введено: in787656908

Результат: in710765611010

### **Вывод.**

Изучены основы обработки символьной информации на языке программирования Ассемблер, получены навыки включать Ассемблерную часть в программу на ЯВУ по принципу встраивания (in-line). Разработана программа, заменяющая в строке десятичные цифры на восьмеричные.

## ПРИЛОЖЕНИЕ А ИСХОДНЫЙ КОД ПРОГРАММ

### Файл lab4.cpp

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

using namespace std;

//глобальные char переменные для использования в __asm
char strIn[81];
char strOut[161];

int main()
{
    FILE* outFile;
    // открытие указанного файла и удаление содержимого
    fopen_s(&outFile, "strOut.txt", "w");

    cout << "Author: Moreva Elena 0382" << endl << "The program for converting decimal digits
to octal" << endl;

    fgets(strIn, 81, stdin); // считывание 80 символов с консоли
    strIn[strlen(strIn) - 1] = '\0';
    __asm {
        push ds
        pop es
        mov esi, offset strIn
        mov edi, offset strOut

        str1 :
            lodsb //загрузить в al из esi

            cmp al, '8'
            jne lb1
            mov ax, '01'
            stosw // ax в edi
            jmp final

        lb1 :

            cmp al, '9'
            jne lb2
            mov ax, '11'
            stosw // ax в edi
            jmp final

        lb2 :

            stosb //запись в строку по адресу edi очередной символ если не число

        final:
            //цикл по строке
            mov ecx, '\0'
            cmp ecx, [esi]
            je strEnd
            jmp str1

        strEnd :
    };
    cout << strOut;
    //запись полученной строки в файл
    fwrite(strOut, sizeof(char), strlen(strOut), outFile);
}
```

```
return 0;  
}
```