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 _____ Подопригора И.П.

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 Nmax (<=80), с клавиатуры в заданную область памяти - на ЯВУ; если длина строки превышает Nmax, остальные символы следует игнорировать;

- выполнение заданного преобразования исходной строки с записью результата в выходную строку - на Ассемблере;

- вывод 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).

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.

Вариант 9:

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

Выполнение работы.

Для выполнения данной лабораторной работы был выбран язык C++ и среда разработки Visual Studio. Блок ассемблерного кода вставлен в программу после ключевого слова __asm. В программе происходит считывание строки длиной не более 80 символов командой fgets, последний символ, который является символом перевода строки, заменяется на завершающий символ (с кодом 0).

В ассемблерном блоке происходит обработка введенной строки, в цикле с помощью команды lodsb считывается очередной символ введенной строки, далее проверяется является ли данный символ ‘8’ или ‘9’ (этих цифр нет в восьмеричной системе счисления) и если он является одной из этих цифр, то в выходную строку с помощью команды stosw заносится 2 символа: “10” или “11” соответственно, иначе символ записывается в выходную строку неизменным с помощью команды stosb. В конце программы полученная выходная строка выводится на экран и в текстовый файл.

Таблица 1. Проверка работы программы.

Введённая строка	Результирующая строка	Примечание
1 2 3 4 5 6 7 8 9 0	1 2 3 4 5 6 7 10 11 0	Верно
Hello888	Hello101010	Верно
98 and 89	1110 and 1011	Верно
Assembler 88909	Assembler 101011011	Верно
Sample text	Sample text	Верно

Тексты исходных файлов программ см. в приложении А.

Выводы.

В ходе данной лабораторной работы была изучена обработка символьной информации с помощью строковых команд на языке ассемблер.

ПРИЛОЖЕНИЕ А

ТЕКСТЫ ИСХОДНЫХ ФАЙЛОВ ПРОГРАММ

Название файла: **lr4.cpp**

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

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

int main()
{
    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, '8'; является ли введённый символ '8'
                  jne skip1
                        mov al, '1'
                        mov ah, '0'
                        stosw
                        jmp final
                  skip1:

                  cmp al, '9'; является ли введённый символ '9'
                  jne skip2
                        mov ah, '1'
                        mov al, '1'
                        stosw
                        jmp final
                        skip2:

                  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;
}
```