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 _____ Позолотин К.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 Nmax (<=80), с клавиатуры в заданную область памяти - на ЯВУ; если длина строки превышает Nmax, 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).

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 19:

Заменить введенные во входной строке латинские буквы на десятичные числа, соответствующие их номеру по алфавиту, остальные символы входной строки передать 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). 3

В ассемблерном блоке происходит обработка введенной строки, в цикле с помощью команды lodsb считывается очередной символ введенной строки, далее проверяется находиться ли данный символ между ‘A’ и ‘Z’, и если это верно в выходную строку с помощью команды stosw заносится 2 символа: “1 ”, “2 ”, “3 ”, “4 ”, “5 ”, …, “10”, “11”, “12”, …, “24”, “25”, “26” 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

Введенная строка	Результирующая строка	Примечание
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26	Верно
чыапфывпфывпфывпр	чыапфывпфывпфывпр	Верно
AaBbCcDdEeFfGgHh	1 a2 b3 c4 d5 e6 f7 g8 h	Верно
AAAkekishBBB	1 1 1 kekish2 2 2	Верно
A B C C B A 123	1 2 3 3 2 1 123	Верно

Тексты исходных файлов программ см. в приложении А.

Выводы.

В ходе выполнения 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>


// A B C D E F G H I J K
L M N O P Q R S T U V W X
Y Z


char s[81];

char outstr[161];


char num[54] = { '1' ,  '
', '2', ' ', '3', ' ',
'4', ' ', '5', ' ',


'6', ' ', '7', ' ', '8',
' ', '9', ' ', '1', '0',


'1', '1', '1', '2', '1',
'3', '1', '4', '1', '5',


'1', '6', '1', '7', '1',
'8', '1', '9', '2', '0',


'2', '1', '2', '2', '2',
'3', '2', '4', '2', '5',


'2', '6', '2', '7'

};


// unsigned char end_str
= '\0';


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
            cmp al, 'A'
            jl skip
            cmp al, 'Z'
            jle replace

            skip :
            stosb
            jmp final

            replace:

            mov ebx, 0
            mov bl, al
            sub ebx, 65
            shl ebx, 1

            mov ah,
[num][ebx + 1]
            mov al,
[num][ebx]

            stosw
            jmp final

            stosw
            jmp final

            final:
        mov  ecx, '\0'
            cmp  ecx,
[esi]
            je   LExit;
```

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