

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. Вариант 18.

Студент гр. 1383

Сардинов М.Г.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2022

Цель работы.

Изучить строковые команды в языке Ассемблер.

Задание.

Заменить введенные во входной строке русские буквы на десятичные числа, соответствующие их номеру по алфавиту, остальные символы входной строки передать в выходную строку непосредственно.

Выполнение работы.

Считывается строка длиной не больше 81 символов после чего записывается в input. Эта строка обрабатывается в ассемблерной вставке, а именно с помощью оператор lodsб в регистр al переходит символ из входной строки после чего определяется русская ли это буква и ее регистр. Если это выполнено то из нее вычитается код начальной буквы ('а' или 'А' в зависимости от регистра) и через некоторые преобразования выводится в выходную строку. Буква ё обрабатывается отдельно.

Тестирование.

Номер	Введённая строка	Выходная строка
1	abcaбвг0098	abc12340098
2	ёг выф 34	74 32922 34

Вывод.

Изучены строковые команда на языке Ассемблер.

ПРИЛОЖЕНИЕ А.

ИСХОДНЫЙ КОД ПРОГРАММЫ

```
#include <iostream>

#include <fstream>

#include <windows.h>

using namespace std;

char input_str[81];

char output_str[165];


int main() {

    SetConsoleCP(1251);

    SetConsoleOutputCP(1251);


    cout << "Сардинов Михаил 1383\nЗадание:\nЗаменить введенные во входной
строке русские буквы на десятичные числа,соответствующие их номеру по
алфавиту,\nпостальные символы входной строки передать в выходную строку
непосредственно.\n";


    cout << "Введите строку:\n";

    cin.getline(input_str, 81);


    ofstream fout("result.txt");

    //cout << (int)('ё') << endl;

    __asm {

        push ds
```

pop es

mov esi, offset input_str

mov edi, offset output_str

start:

lodsb

mov ah, 0

cmp al, 'A'

JGE nextB1

jmp fin1

nextB1:

cmp al, 'Я'

JLE nextB2

jmp fin1

nextB2:

sub al, 'A'

jmp chang

fin1:

cmp al, 'a'

JGE nextS1

jmp nextS3

nextS1:

 cmp al, 'я'

 JLE nextS2

 jmp nextS3

nextS2:

 sub al, 'a'

 jmp chang

jmp nextS3

chang:

 cmp al, 7

 JG some1

 JLE some2

some1:

 add al,2

 mov bl,10

 div bl

 add al, '0'

 stosb

 mov al,ah

 add al, '0'

 stosb

jmp start

some2:

cmp al, 5

JG some3

jmp nextn

some3:

add al, 1

nextn:

add al, '1'

stosb

jmp start

nextS3:

cmp al,'Ë'

JE change2

cmp al, 'ë'

JE change2

jmp nextn1

change2:

mov al, '7'

nextn1:

stosb

cmp[esi], '\0'

```
jne start
```

```
};
```

```
cout << output_str;
```

```
fout << output_str;
```

```
fout.close();
```

```
return 0;
```

```
}
```

