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 символьных команд»

Студент гр. 1383 _____ Петров А.С.

Преподаватель ____ _____ Ефремов М. А.

Санкт-Петербург

2022

Цель работы.

Изучить представление и обработку символьной информации с использованием символьных команд.

Задание на лабораторную работу.

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).

Вариант 16

Преобразование введенных во входной строке русских букв в латинские в соответствие с правилами транслитерации, остальные символы входной строки передаются в выходную строку непосредственно.

Выполнение работы.

На ЯВУ происходит смена кодировки на windows 1251, после чего происходит считывание строки символов, введенных с клавиатуры. Затем начинается модуль программы, написанный на языке Ассемблера, где при помощи команды lodsb происходит запись символа из входной строки в регистр al. После этого проверяется является ли этот символ русским, если является, то происходит его транслитерация при помощи изменения значения в регистре al на необходимое. Затем происходит запись полученного символа или символов в выходную строку при помощи команды stosb и программа начинает обработку следующего символа из входной строки. Если символ не является русским, то в выходную строку он записывается без изменений. Процесс длится до того момента, пока не в регистр al не будет записан 0 (символ ‘\0’ – конец строки). После преобразований на языке Ассемблера новая строка выводится на экран и записывается в файл “out.txt”.

Тестирование

Результаты тестирования представлены в таблице 1.

Таблица 1 – Результаты тестирования.

Номер теста	Входные данные	Результат	Ожидаемый результат
1	Jack Лондон ест рыбу	Jack London est rybu	Jack London est rybu
2	Сегодня ужасная погода	Segodnia ujasnaia pogoda	Segodnia ujasnaia pogoda
3	fish. В МЕТРОПОЛИТЕНЕ ТЕПЛО. Люди ловят рыбу. gun,	fish. V METROPOLITENE TEPLO. Liudi loviat rybu. gun,	fish. V METROPOLITENE TEPLO. Liudi loviat rybu. gun,

Выводы.

В ходе выполнения работы были изучены представление символьной информации, обработка символьной информации и символьные команды.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: lr4.c

```
#include <stdio.h>
int main(){
    system("chcp 1251 > nul");
    printf     ("Петров     Александр\nГруппа     1383\nВариант
16\nПреобразование введенных во входной строке русских букв в
латинские в соответствие с правилами транслитерации, остальные
символы входной строки передаются в выходную строку
непосредственно.\n");
    char input[81];
    char output[81];
    fgets(input,81,stdin);
    asm(

        "letter_check: \n"
            "lodsb \n"
            "cmp al, 0\n"
            "je finish \n"

            "cmp al, 192\n"//AAAAA
            "je A_sym \n"
            "cmp al, 224\n"//aaaaaa
            "je a_sym \n"

            "cmp al, 193\n"//БББББ
            "je B_sym \n"
            "cmp al, 225\n"//ббббб
            "je b_sym \n"

            "cmp al, 194\n"//ВВВВВ
            "je V_sym \n"
            "cmp al, 226\n"//ввввв
            "je v_sym \n"

            "cmp al, 195\n"//ГГГГГ
            "je G_sym \n"
            "cmp al, 227\n"//ггггг
            "je g_sym \n"

            "cmp al, 196\n"//ДДДДД
            "je D_sym \n"
            "cmp al, 228\n"//ддддд
            "je d_sym \n"

            "cmp al, 197\n"//ЕЕЕЕЕ
            "je E_sym \n"
            "cmp al, 229\n"//eeeee
```

```
"je e_sym \n"

"cmp al, 198\n"//ЖЖЖЖЖ
"je J_sym \n"
"cmp al, 230\n"//жжжжж
"je j_sym \n"

"cmp al, 199\n"//ЗЗЗЗЗ
"je Z_sym \n"
"cmp al, 231\n"//ззззз
"je z_sym \n"

"cmp al, 200\n"//ИИИИИ
"je I_sym \n"
"cmp al, 232\n"//ииииии
"je i_sym \n"

"cmp al, 201\n"//ЙЙЙЙЙ
"je I_sym \n"
"cmp al, 233\n"//ййййй
"je i_sym \n"

"cmp al, 202\n"//ККККК
"je K_sym \n"
"cmp al, 234\n"//ккккк
"je k_sym \n"

"cmp al, 203\n"//ЛЛЛЛЛ
"je L_sym \n"
"cmp al, 235\n"//лллллл
"je l_sym \n"

"cmp al, 204\n"//МММММ
"je M_sym \n"
"cmp al, 236\n"//ммммм
"je m_sym \n"

"cmp al, 205\n"//ННННН
"je N_sym \n"
"cmp al, 237\n"//ннннн
"je n_sym \n"

"cmp al, 206\n"//ООООО
"je O_sym \n"
"cmp al, 238\n"//ооооо
"je o_sym \n"

"cmp al, 207\n"//ППППП
"je P_sym \n"
"cmp al, 239\n"//ппппп
"je p_sym \n"

"cmp al, 208\n"//PPPPP
```

```
"je R_sym \n"
"cmp al, 240\n"//ppppp
"je r_sym \n"

"cmp al, 209\n"//CCCCC
"je S_sym \n"
"cmp al, 241\n"//ccccc
"je s_sym \n"

"cmp al, 210\n"//TTTTT
"je T_sym \n"
"cmp al, 242\n"//ттттт
"je t_sym \n"

"cmp al, 211\n"//УУУУУ
"je U_sym \n"
"cmp al, 243\n"//ууууу
"je u_sym \n"

"cmp al, 212\n"//ФФФФФ
"je F_sym \n"
"cmp al, 244\n"//ффффф
"je f_sym \n"

"cmp al, 213\n"//XXXXX
"je Kh_sym \n"
"cmp al, 245\n"//xxxxx
"je kh_sym \n"

"cmp al, 214\n"//ЦЦЦЦЦ
"je C_sym \n"
"cmp al, 246\n"//ццццц
"je c_sym \n"

"cmp al, 215\n"//ЧЧЧЧЧ
"je Ch_sym \n"
"cmp al, 247\n"//ччччч
"je ch_sym \n"

"cmp al, 216\n"//ШШШШШ
"je Sh_sym \n"
"cmp al, 248\n"//шшшшш
"je sh_sym \n"

"cmp al, 217\n"//ЩЩЩЩЩ
"je Sc_sym \n"
"cmp al, 249\n"//щщщщщ
"je sc_sym \n"

"cmp al, 218\n"//ЬЬЬЬ
"je letter_check \n"
"cmp al, 250\n"//ььььь
"je letter_check \n"
```

```
    "cmp al, 219\n"//ЫЫЫЫ
    "je Y_sym \n"
    "cmp al, 251\n"//ыыыыы
    "je y_sym \n"

    "cmp al, 220\n"//ЪЪЪЪ
    "je letter_check \n"
    "cmp al, 252\n"//ъъъъъ
    "je letter_check \n"

    "cmp al, 221\n"//ЭЭЭЭЭ
    "je E_sym \n"
    "cmp al, 253\n"//эээээ
    "je e_sym \n"

    "cmp al, 222\n"//ЮЮЮЮЮ
    "je Iu_sym \n"
    "cmp al, 254\n"//ююююю
    "je iu_sym \n"

    "cmp al, 223\n"//ЯЯЯЯЯ
    "je Ia_sym \n"
    "cmp al, 255\n"//яяяяя
    "je ia_sym \n"

    "stosb \n"
    "jmp letter_check \n"

"A_sym: \n"//A
    "mov al, 65 \n"
    "stosb \n"
    "jmp letter_check \n"

"a_sym: \n"//a
    "mov al, 97 \n"
    "stosb \n"
    "jmp letter_check \n"

"B_sym: \n"//B
    "mov al, 66 \n"
    "stosb \n"
    "jmp letter_check \n"

"b_sym: \n"//b
    "mov al, 98 \n"
    "stosb \n"
    "jmp letter_check \n"

"C_sym: \n"//C
    "mov al, 67 \n"
    "stosb \n"
    "jmp letter_check \n"
```

```
"c_sym: \n"//c
    "mov al, 99 \n"
    "stosb \n"
    "jmp letter_check \n"

"D_sym: \n"//D
    "mov al, 68 \n"
    "stosb \n"
    "jmp letter_check \n"

"d_sym: \n"//d
    "mov al, 100 \n"
    "stosb \n"
    "jmp letter_check \n"

"E_sym: \n"//E
    "mov al, 69 \n"
    "stosb \n"
    "jmp letter_check \n"

"e_sym: \n"//e
    "mov al, 101 \n"
    "stosb \n"
    "jmp letter_check \n"

"F_sym: \n"//F
    "mov al, 70 \n"
    "stosb \n"
    "jmp letter_check \n"

"f_sym: \n"//f
    "mov al, 102 \n"
    "stosb \n"
    "jmp letter_check \n"

"G_sym: \n"//G
    "mov al, 71 \n"
    "stosb \n"
    "jmp letter_check \n"

"g_sym: \n"//g
    "mov al, 103 \n"
    "stosb \n"
    "jmp letter_check \n"

"H_sym: \n"//H
    "mov al, 72 \n"
    "stosb \n"
    "jmp letter_check \n"

"h_sym: \n"//h
    "mov al, 104 \n"
```

```
    "stosb \n"
    "jmp letter_check \n"

"I_sym: \n"//I
    "mov al, 73 \n"
    "stosb \n"
    "jmp letter_check \n"

"i_sym: \n"//i
    "mov al, 105 \n"
    "stosb \n"
    "jmp letter_check \n"

"J_sym: \n"//J
    "mov al, 74 \n"
    "stosb \n"
    "jmp letter_check \n"

"j_sym: \n"//j
    "mov al, 106 \n"
    "stosb \n"
    "jmp letter_check \n"

"K_sym: \n"//K
    "mov al, 75 \n"
    "stosb \n"
    "jmp letter_check \n"

"k_sym: \n"//k
    "mov al, 107 \n"
    "stosb \n"
    "jmp letter_check \n"

"L_sym: \n"//L
    "mov al, 76 \n"
    "stosb \n"
    "jmp letter_check \n"

"l_sym: \n"//l
    "mov al, 108 \n"
    "stosb \n"
    "jmp letter_check \n"

"M_sym: \n"//M
    "mov al, 77 \n"
    "stosb \n"
    "jmp letter_check \n"

"m_sym: \n"//m
    "mov al, 109 \n"
    "stosb \n"
    "jmp letter_check \n"
```

```
"N_sym: \n"//N
    "mov al, 78 \n"
    "stosb \n"
    "jmp letter_check \n"

"n_sym: \n"//n
    "mov al, 110 \n"
    "stosb \n"
    "jmp letter_check \n"

"O_sym: \n"//O
    "mov al, 79 \n"
    "stosb \n"
    "jmp letter_check \n"

"o_sym: \n"//o
    "mov al, 111 \n"
    "stosb \n"
    "jmp letter_check \n"

"P_sym: \n"//P
    "mov al, 80 \n"
    "stosb \n"
    "jmp letter_check \n"

"p_sym: \n"//p
    "mov al, 112 \n"
    "stosb \n"
    "jmp letter_check \n"

"R_sym: \n"//R
    "mov al, 82 \n"
    "stosb \n"
    "jmp letter_check \n"

"r_sym: \n"//r
    "mov al, 114 \n"
    "stosb \n"
    "jmp letter_check \n"

"S_sym: \n"//S
    "mov al, 83 \n"
    "stosb \n"
    "jmp letter_check \n"

"s_sym: \n"//s
    "mov al, 115 \n"
    "stosb \n"
    "jmp letter_check \n"

"T_sym: \n"//T
    "mov al, 84 \n"
    "stosb \n"
```

```
    "jmp letter_check \n"

"t_sym: \n"//t
    "mov al, 116 \n"
    "stosb \n"
    "jmp letter_check \n"

"U_sym: \n"//U
    "mov al, 85 \n"
    "stosb \n"
    "jmp letter_check \n"

"u_sym: \n"//u
    "mov al, 117 \n"
    "stosb \n"
    "jmp letter_check \n"

"V_sym: \n"//V
    "mov al, 86 \n"
    "stosb \n"
    "jmp letter_check \n"

"v_sym: \n"//v
    "mov al, 118 \n"
    "stosb \n"
    "jmp letter_check \n"

"Y_sym: \n"//Y
    "mov al, 89 \n"
    "stosb \n"
    "jmp letter_check \n"

"y_sym: \n"//y
    "mov al, 121 \n"
    "stosb \n"
    "jmp letter_check \n"

"Z_sym: \n"//Z
    "mov al, 90 \n"
    "stosb \n"
    "jmp letter_check \n"

"z_sym: \n"//z
    "mov al, 122 \n"
    "stosb \n"
    "jmp letter_check \n"

"Kh_sym: \n"//Kh
    "mov al, 75 \n"
    "stosb \n"
    "mov al, 104 \n"
    "stosb \n"
    "jmp letter_check \n"
```

```
"kh_sym: \n"//kh
    "mov al, 107 \n"
    "stosb \n"
    "mov al, 104 \n"
    "stosb \n"
    "jmp letter_check \n"

"Ch_sym: \n"//Ch
    "mov al, 67 \n"
    "stosb \n"
    "mov al, 104 \n"
    "stosb \n"
    "jmp letter_check \n"

"ch_sym: \n"//ch
    "mov al, 99 \n"
    "stosb \n"
    "mov al, 104 \n"
    "stosb \n"
    "jmp letter_check \n"

"Sh_sym: \n"//Sh
    "mov al, 83 \n"
    "stosb \n"
    "mov al, 104 \n"
    "stosb \n"
    "jmp letter_check \n"

"sh_sym: \n"//sh
    "mov al, 115 \n"
    "stosb \n"
    "mov al, 104 \n"
    "stosb \n"
    "jmp letter_check \n"

"Sc_sym: \n"//Sc
    "mov al, 83 \n"
    "stosb \n"
    "mov al, 99 \n"
    "stosb \n"
    "jmp letter_check \n"

"sc_sym: \n"//sc
    "mov al, 115 \n"
    "stosb \n"
    "mov al, 99 \n"
    "stosb \n"
    "jmp letter_check \n"

"Iu_sym: \n"//Iu
    "mov al, 73 \n"
    "stosb \n"
```

```
            "mov al, 117 \n"
            "stosb \n"
            "jmp letter_check \n"

        "iu_sym: \n"//iu
            "mov al, 105 \n"
            "stosb \n"
            "mov al, 117 \n"
            "stosb \n"
            "jmp letter_check \n"

        "Ia_sym: \n"//Ia
            "mov al, 73 \n"
            "stosb \n"
            "mov al, 97 \n"
            "stosb \n"
            "jmp letter_check \n"

        "ia_sym: \n"//ia
            "mov al, 105 \n"
            "stosb \n"
            "mov al, 97 \n"
            "stosb \n"
            "jmp letter_check \n"

        "finish: \n"
            "stosb \n"

        :
        : "D"(output), "S"(input)
        :
    );



    printf("%s\n", &output);
    FILE * file = fopen ("out.txt", "w");
    fprintf(file, "%s", output);
    fclose (file);
};
```