МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра МОЭВМ

отчет

по лабораторной работе №4

по дисциплине «Организация ЭВМ и систем»

Тема: Представление и обработка символьной информации с использованием строковых команд

Студент гр. 9382 _____ Рыжих Р.В.

Преподаватель _____ Ефремов М.А.

Санкт-Петербург

2020

Цель работы.

Научиться обрабатывать символьную информацию с использованием строковых команд.

Основные теоретические положения.

Задание:

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 17

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

Ход работы:

Происходит инициализация массивов для ввода и вывода на языке C++, в дальнейшем идет ассемблерная вставка _asm.

В ассемблерной вставке мы инициализируем регистры si и di так, чтобы они отвечали за строки ввода и вывода. Дальше, мы считываем символ из строки с помощью команды lodsb, а затем проверяем, какой этот символ, и печатаем

транслитный ему символ с помощью команды stosb (при вводе некоторых строк, например: ch, zh, sch, sh, ya, yu, yo – будут выводится в новую строку ч, ж, щ, ш, я, ю, ё)

После печати мы заново считываем символ из входной строки и повторяем все действия.

В завершении программа выводит на экран обработанную строку.

Тестирование.

№	Входные данные	Выходные данные
1	zhaba dushit golovnyak	жаба душит головняк
2	cherezva	черезва
3	ch zh sh sch ya yo yu	ч ж ш щ я ё ю
4	chervyachok dzhimm	червячок джимм

Выводы.

В результате выполнения лабораторной работы была изучена обработка символьной информации с использованием строковых команд.

ПРИЛОЖЕНИЕ А
ИСХОДНЫЙ КОД ПРОГРАММЫ

```
#include <iostream>
#include <stdio.h>
#include <windows.h>
#define N 80
using namespace std;

int main() {

    system("chcp 1251 > nul");
    cout << "Лабораторная работа номер 4, вариант 17\n"
        "Преобразование введенных во входной строке латинских букв в русские в соответ-
ствие\n"
        "с правилами транслитерации, остальные символы входной строки передаются в выход-
ную\n"
        "строку непосредственно. \n";
        "Выполнил Рыжих Р.В. группа 9382\n";
    char input[N+1];
    cin.getline(input, N+1);
    char output[N+1] = "";
    _asm {
        lea si, input
        lea di, output
        cld
        replacer:
            lodsb
        replacer_1:
            cmp al, 'a'; Если < a не буква
            jl WriteSymbol;

            cmp al, 'z'; Если > z не буква
            jg WriteSymbol;

            cmp al, 'a'
            je replace_a

            cmp al, 'b'
                je replace_b

            cmp al, 'c'
                je replace_c

            cmp al, 'd'
                je replace_d

            cmp al, 'e'
                je replace_e

            cmp al, 'f'
                je replace_f

            cmp al, 'g'
                je replace_g

            cmp al, 'h'
                je replace_h

            cmp al, 'i'
```

```
    je replace_i

cmp al, 'j'
    je replace_j

cmp al, 'k'
    je replace_k

cmp al, 'l'
    je replace_l

cmp al, 'm'
    je replace_m

cmp al, 'n'
    je replace_n

cmp al, 'o'
    je replace_o

cmp al, 'p'
    je replace_p

cmp al, 'q'
    je replace_q

cmp al, 'r'
    je replace_r

cmp al, 's'
    je replace_s

cmp al, 't'
    je replace_t

cmp al, 'u'
    je replace_u

cmp al, 'v'
    je replace_v

cmp al, 'w'
    je replace_w

cmp al, 'x'
    je replace_x

cmp al, 'y'
    je replace_y

cmp al, 'z'
    je replace_z

    jmp WriteSymbol

        replace_a:
    mov al, 'a'
        stosb
        jmp replacer

        replace_b :
    mov al, 'б'
        stosb
        jmp replacer
```

```
    replace_c :
    lodsb
    cmp al, 'h'
    je replace_ch
    mov bl, al
    mov al, 'ц'
    stosb
    mov al, bl
    jmp replacer_1

    replace_ch:
mov al, 'ч'
    stosb
    jmp replacer

    replace_d :
mov al, 'д'
    stosb
    jmp replacer

    replace_e :
mov al, 'е'
    stosb
    jmp replacer

    replace_f :
mov al, 'ф'
    stosb
    jmp replacer

    replace_g :
mov al, 'г'
    stosb
    jmp replacer

    replace_h :
mov al, 'х'
    stosb
    jmp replacer

    replace_i :
mov al, 'и'
    stosb
    jmp replacer

    replace_j :
mov al, 'й'
    stosb
    jmp replacer

    replace_k :
mov al, 'к'
    stosb
    jmp replacer

    replace_l :
mov al, 'л'
    stosb
    jmp replacer

    replace_m :
mov al, 'м'
    stosb
    jmp replacer
```

```
    replace_n :
mov al, 'н'
    stosb
    jmp replacer

    replace_o :
mov al, 'о'
    stosb
    jmp replacer

    replace_p :
mov al, 'п'
    stosb
    jmp replacer

    replace_q :
    mov al, 'к'
    stosb
    mov al, 'у'
    stosb
        jmp replacer

    replace_r :
mov al, 'р'
    stosb
    jmp replacer

    replace_s :
    lodsb
    cmp al, 'с'
    je replace_sc
    cmp al, 'h'
    je replace_sh
    mov bl, al
    mov al, 'с'
    stosb
    mov al, bl
    jmp replacer_1

    replace_sc:
lodsb
    mov bl, al
    cmp al, 'h'
    je replace_sch
    mov al, 'с'
    stosb
    mov al, 'ц'
    stosb
    mov al, bl
    jmp replacer_1

    replace_sch:
mov al, 'щ'
    stosb
    jmp replacer

    replace_sh:
mov al, 'ш'
    stosb
    jmp replacer

    replace_t :
mov al, 'т'
    stosb
    jmp replacer
```

```
    replace_u :
mov al, 'у'
    stosb
    jmp replacer

    replace_v :
mov al, 'в'
    stosb
    jmp replacer

    replace_w :
mov al, 'в'
    stosb
    jmp replacer

    replace_x :
mov al, 'х'
    stosb
    jmp replacer

    replace_y :
lodsb
    cmp al, 'a'
    je replace_ya
    cmp al, 'u'
    je replace_yu
    cmp al, 'o'
    je replace_yo
    mov bl, al
    mov al, 'и'
    stosb
    mov al, bl
    jmp replacer_1

    replace_ya:
mov al, 'я'
    stosb
    jmp replacer

    replace_yu :
mov al, 'ю'
    stosb
    jmp replacer

    replace_yo :
mov al, 'ё'
    stosb
    jmp replacer

    replace_z :
lodsb
    cmp al, 'h'
    je replace_zh
    mov bl, al
    mov al, 'з'
    stosb
    mov al, bl
    jmp replacer_1

    replace_zh:
mov al, 'ж'
    stosb
    jmp replacer
```

```
            WriteSymbol:
                stosb
                    dec ecx
                    jns replacer

                    mov al, 0

    }
    cout << output;
    return 0;
}
```