**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 нa 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 С++, 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 соответствие\n"

"с правилами транслитерации, остальные символы входной строки передаются в выходную\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, 'а'

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, 'c'

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;

}