

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 15.

Студент гр. 0382

Санников В.А.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2021

Цель работы.

Изучить представление и обработку символьной информации с использованием строковых команд на языке Ассемблера. Разработать программу, которая обрабатывает строку.

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 N_{\max} (≤ 80), с клавиатуры в заданную область памяти - на ЯВУ; если длина строки превышает N_{\max} , 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 №15: Исключение русских букв и цифр, введенных во входной строке, при формировании выходной строки.

Ход работы:

Выполнение лабораторной работы проводилось на языке C++ MCVS.

В начале программы в консоль выводится строка с именем, номером группы и заданием. Далее с помощью getline считывается 81 символ с потока (или меньше).

Setlocale и system дают нам возможность работать с кириллицей.

Далее объявляется ассемблерный блок через __asm, в котором происходит считывание каждого символа введенной строки с помощью lodsб (помещает байт из es:esi в al). В процессе жизни программа проверяет

каждый символ через промежутки '0'-'9' и 'A'-'я' (не забываем обработать исключение 'Ё' и 'ё'). Если символ выходит за эти границы, то его программа игнорирует (в этом нам помогают метки и команды перехода к меткам: je, jg, jl, jmp). Если же символ подходит под условие, то по команде stosb, его байт al записывается в es:edi. Если встречается символ конца строки, ассемблерный блок заканчивается.

В конце, полученная строка на ЯВУ выводится на экран и записывается в текстовый файл.

Исходный код программы см. в приложении А.

Тестирование:

Для проверки работоспособности программы были проведены тесты, см. Таблицу 1.

Таблица 1 — Результаты тестирования.

№ теста	Входные данные	Выходные данные	Оценка результата
1	АААяяяа12345678 90		Верно
2	_Hello Seattle!_	_Hello Seattle!_	Корректно
3	Привет за ов1раг3ом prileg дед0 >:30	V prileg >:	Верно

Выводы.

В результате лабораторной работы была изучена обработка символьной информации с использованием ассемблерного блока в коде на ЯВУ.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: main.cpp

```
#include <iostream>
#include <fstream>

using namespace std;

char input[81];
char output[81];

int main(){
    system("chcp 1251 > nul");
    setlocale(LC_CTYPE, "rus");
    cout << "Sannikov Vadim 0382, task: delete russian symbols
and digits" << endl;
    ofstream file;
    file.open(R"(C:\Users\79081\source\repos\lab_4\lab_4\
out.txt)");
    cin.getline(input, 81);
    __asm {
        mov esi, offset input
        mov edi, offset output

        check_symb:
            lodsb
            cmp al, '\0'
            je finish

            cmp al, 'ë'
            je check_symb
            cmp al, 'Ё'
            je check_symb
            cmp al, 'А'
            jl is_digit
            cmp al, 'я'
            jg is_digit
            jmp check_symb

        is_digit:
            cmp al, '0'
            jl write_symb
            cmp al, '9'
            jg write_symb
            jmp check_symb

        write_symb:
            stosb
            jmp check_symb
    }
```

```
        finish:
    };
    cout << output;
    file << output;
    file.close();
    return 0;
}
```