

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

Русинов Д.А.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2020

Цель работы.

Научиться представлять и обрабатывать символьную информацию и изучить встраивание in-line

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 5. Преобразование всех строчных латинских букв входной строки в заглавные, а десятичных цифр в инверсные, остальные символы входной строки передаются в выходную строку непосредственно.

Выполнение работы.

Создаются два указателя – входная строка и выходная строка. Адреса данных строк записываются в регистры `rsi` и `rdi`. Затем проходимся по всей строке

Тестирование.

Номер	Входные данные	Выходные данные
1	ASSembler123	ASSEMBLER876
2	asd a 1 0 asd 9 AAAA прив	ASD A 8 9 ASD 0 AAAA прив
3	sorry but sorry for what	SORRY BUT SORRY FOR WHAT
4	My DaaddYY1 tolds me2	MY DAADDYY8 TOLDS ME7

Выводы.

Была изучена обработка символьной информации, освоен принцип встраивания in-line

ПРИЛОЖЕНИЕ А

Название файла: main.cpp

```
#include <iostream>

int main()
{
    std::cout << "Работу выполнил: студент группы 9382 Русинов Дмитрий"
    << std::endl;
    std::cout << "Вид преобразования: 5. Преобразование всех строчных ла-
    тинских букв входной строки в" << std::endl;
    std::cout << "заглавные, а десятичных цифр в инверсные, остальные
    символы входной строки" << std::endl;
    std::cout << "передаются в выходную строку непосредственно." <<
    std::endl;

    char *input = new char[80];
    char *output = new char[80];

    std::cout<<"Входная строка: ";
    std::cin.getline(input, 80);

    asm("mov  %0,%%rsi\n\t"
        "mov  %1,%%rdi\n\t"
        "mov  $80,%%ecx\n\t"

        "start:"
        "lods b (%rsi)\n\t" //загружаем символ в al
        "cmp  $0x30,%%al\n\t" //сравниваем символ с кодом цифры 0
        "j l character\n\t" //если меньше, то не цифра, идем дальше к
        проверке на буквы
        "cmp  $0x39,%%al\n\t" //сравниваем символ с кодом цифры 9
        "j g character\n\t" //если больше то не цифра в 10 сс идем к
        проверке на буквы
        "sub  $0x39,%%al\n\t" //вычитаем 39 чтобы получить цифру [-
        9...0]
        "neg  %%al\n\t" //берем модуль числа и получаем инверти-
        рованную цифру
        "add  $0x30,%%al\n\t" //прибавляем 30 чтобы получить код цифры
        "j mp print\n\t" //переходим к выводу в выходную строку

        "character:"
        "cmp  $0x61,%%al\n\t" //сравниваем с символом "a"
        "j l print\n\t" //если меньше, переходим к выводу в вы-
        ходную строку
        "cmp  $0x7a,%%al\n\t" //сравниваем с символом "z"
        "j g print\n\t" //если больше, то преходим к выводу в вы-
        ходную строку
        "sub  $0x20,%%al\n\t" //получаем заглавную букву

        "print:"
        "stos b (%rdi)\n\t" //записываем символ в выходную строку
        "loop start\n\t" //возвращаемся в start пока ecx!=0
        : "a"(input), "b"(output));

    std::cout<<"Выходная строка: " << output << std::endl;
```

```
    delete[] input;  
    delete[] output;  
  
    return 0;  
}
```