

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

Студент(ка) гр. 9382

Голубева В.П.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2020

Цель работы.

Научиться представлять и обрабатывать символьную информацию, и изучить принцип встраивания 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 4

Преобразование всех заглавных латинских букв входной строки в строчные, а восьмеричных цифр в инверсные, остальные символы входной строки передаются в выходную строку непосредственно.

Выполнение работы.

Создаём два указателя на строки. Один из них нужен для записи входной строки, другой для записи выходной строки, затем проходимся по символам этой строки и записываем преобразованные по необходимости символы.

Тестирование.

Тестирование представлено в таблице 1.

Таблица 1. Тестирование программы

Номер	Входные данные	Выходные данные
1	jhcj 9 HELLO ораор редств	hcj 9 hello ораор редств
2	sdfs2165HH	sdfs5612hh
3	jhj	jhj
4	AbCdEfGh1234567	abcdefgh6543210

Выводы.

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

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: lab4_1.cpp

```
#include <iostream>
```

```
#include <fstream>
```

```
int main()
```

```
{
```

```
std::cout<<"
```

```
_____ \n";
```

```
std::cout<<"| Вид преобразования: 4. Преобразование всех  
заглавных латинских букв входной строки в | \n";
```

```
std::cout<<"| строчные, а восьмеричных цифр в инверсные,  
остальные символы входной строки | \n";
```

```
std::cout<<"| передаются в выходную строку непосредственно.  
| \n";
```

```
std::cout<<"| Работу выполнила: студентка группы 9382  
Голубева Валентина | \n";
```

```
std::cout<<"
```

```
_____ \n";
```

```
char *str1 = new char[80];
```

```
std::cout<<"Введите строку: ";
```

```
std::cin.getline(str1,80);
```

```
char *str2 = new char[80];
```

```
asm("mov %0,%%rsi\n\t" //положили в al 0
```

```
"mov %1,%%rdi\n\t"
```

```
"mov $80,%%ecx\n\t" //положили длину строки в ecx
```

```
"beginning:"
```

```
"lodsb (%%rsi)\n\t" //загружаем символ в al
```

```
"cmpb $0x30,%%al\n\t" //сравниваем символ с кодом цифры
```

0

```

        "j1 character\n\t" //если меньше, то не цифра, идем
дальше к проверке на буквы
        "cmpb $0x37,%%al\n\t" //сравниваем символ с кодом цифры
7
        "jg character\n\t" //если больше то не цифра в 8 сс
идем к проверке на буквы
        "sub $0x30,%%al\n\t" //вычитаем 30 чтобы получить
цифру
        "xor $0x7,%%al\n\t" //инвертируем последние 3 бита
        "add $0x30,%%al\n\t" //прибавляем 30 чтобы получить
код цифры
        "jmp print\n\t" //переходим к выводу в выходную
строку

        "character:"
        "cmpb $0x41,%%al\n\t " //сравниваем с символом "A"
        "jl print\n\t" //если меньше, переходим к выводу
в выходную строку
        "cmpb $0x5a,%%al\n\t" //сравниваем с символом "Z"
        "jg print\n\t" //если больше, то преходим к
выводу в выходную строку
        "add $0x20,%%al\n\t" //получаем строчную букву

        "print:"
        "stosb (%rdi)\n\t" //записываем символ в выходную
строку
        "loop beginning\n\t" //возвращаемся в начало пока есх!
=0
        ::"m"(str1),"m"(str2)
    );

    std::cout<<"Преобразованная строка:\t"<<str2<<'\n';
    std::ofstream fileOut("Result.txt");
    if(fileOut.is_open())
    {
        fileOut<<str2;
        fileOut.close();

```

```
    }  
  
    delete[] str1;  
    delete[] str2;  
  
    return 0;  
}
```