

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

Студентка гр. 9383

Лихашва А.Д.

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 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 9

Преобразование введенных во входной строке десятичных цифр в восьмеричную СС, 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 `_asm` в зависимости от длины исходной строки настраиваются регистры, отвечающие за индексы, то есть происходит подготовка к циклу `traverse`.

При помощи команды `cmp` происходит сравнение элементов массива с цифрами «8» и «9», потому что именно эти цифры в восьмеричной СС отличаются от десятичной СС, остальные цифры одинаковые. Если находятся цифры, равны «8» или «9», тогда мы прыгаем на соответствующие метки, и меняем их значение под восьмеричную СС. Если после обработки остаются еще элементы, то мы заново повторяем цикл `traverse`.

В завершении программа выводит на экран преобразованную строку.

Тестирование.

№	Входные данные	Выходные данные
1	7838734DF	710310734DF
2	999	111111
3	878348735A12S	107103410735A12S
4	88899	1010101111

Выводы.

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

Файл `lab4.cpp` находится в приложении А.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

```
#include <iostream>
#include <stdio.h>

#define N 80
using namespace std;

int main()
{
    setlocale(LC_ALL, "Russian");
    cout << "
_____|\\n";
    cout << "|
|\\n";
    cout << "|Лабораторная работа №4. Вариант 9
|\\n";
    cout << "|Выполнила Лихашва Анастасия, группа 9383
|\\n";
    cout << "|Преобразование введенных во входной строке шестна-
дцатиричных цифр в десятичную СС, |\\n";
    cout << "|остальные символы входной строки передаются в выход-
ную строку непосредственно. |\\n";
    cout << "|
_____|\\n\\n";
    cout << "Введите строку:\\n";

    char str_in[N];
    char str_out[N * 2];
    int i = 0;
    cin.getline(str_in, N);
    _asm {
        sub eax, eax;
        mov al, 0          //al - символ конца строки
        mov ecx, N          //ecx = N
        lea edi, str_in     //edi указывает на начало _str
        repne scas          //ecx содержит N - str.length (длина
строки)

        sub ecx, N          //ecx = -str.length
        neg ecx             //ecx = str.length
        mov edx, ecx        //edx = ecx
        sub edi, edi        //edi == 0
        sub esi, esi        //esi == 0

        traverse:           //обход
```

```

        mov edi, edx      //edi = edx
        sub edi, ecx //edi - указывает на последний элемент в
str; когда вычитается ecx, то указывается на текущий индекс, так
как ecx уменьшается с каждой итерацией

        mov al, str_in[edi]    //al содержит текущий элемент

        cmp al, '8' //если равен 8
        je eight8

        cmp al, '9' //если равен 9
        je eight9

        jmp write

eight8 :    //8 меняется на 10
mov str_out[esi], '1'
inc esi //увеличивается значение в регистре ESI на 1
mov str_out[esi], '0'
inc esi //увеличивается значение в регистре ESI на 1
loop traverse //цикл

eight9 :    //9 меняется на 11
mov str_out[esi], '1'
inc esi //увеличивается значение в регистре ESI на 1
mov str_out[esi], '1'
inc esi //увеличивается значение в регистре ESI на 1
loop traverse //цикл

write :
mov str_out[esi], al //заносим al
inc esi //увеличивается значение в регистре ESI на 1
loop traverse //цикл

mov str_out[esi], 0
}
cout << "Преобразованная строка:\n" << str_out;
return 0;
}

```