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 _____ Чебесова И.Д.

Преподаватель _____ Ефремов М.А.

Санкт-Петербург
2020

Цель работы.

Изучить представление и обработку символьной информации — строки символов на языке Ассемблер.

Написать программу, обрабатывающую строку по определенному принципу на языке высокого уровня (C++) с включением фрагмента на языке Ассемблер по принципу встраивания (in-line).

Текст задания.

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 на ЯВУ по принципу встраивания (in-line).

Исходные данные.

Вариант 22

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

Ход работы.

В ходе работы была разработана программа на языке C++ и Ассемблер, которая заменяет все латински заглавные буквы на строчные, а десятичные цифры инверсирует.

Замена заглавных букв происходит по принципу смещения в таблице аски на 20, так как закглавная буква «A» имеет номер 0x41, когда как прописная «a» имеет номер 0x61, остальные же буквы идут дальше по порядку без перерывов.

Инверсия цифр происходит по следующей схеме: сначала из номера цифры в таблице вычитается номер 9 — 0x39. Таким образом мы получаем тем самым инверсированную цифру, но со знаком «-», который мы убираем операцией *neg*, затем прибавляем обратно 0x30, получая нужную нам цифру.

Таблица 1 — объяснение преобразования цифр.

Цифра	Инверсная цифра	Код цифры в ASCII	Преобразование
0	9	0x30	- (0x30 — 0x39) = 9
1	8	0x31	- (0x31 — 0x39) = 8
2	7	0x32	- (0x32 — 0x39) = 7
3	6	0x33	- (0x33 — 0x39) = 6
4	5	0x34	- (0x34 — 0x39) = 5
5	4	0x35	- (0x35 — 0x39) = 4
6	3	0x36	- (0x36 — 0x39) = 3
7	2	0x37	- (0x37 — 0x39) = 2
8	1	0x38	- (0x38 — 0x39) = 1
9	0	0x39	- (0x39 — 0x39) = 0

В файле *lb4.cpp* находится три функции:

void printINFO() - она выводит на экран информацию об авторе и преобразованию.

char change(char* str_in)* — функция, которая принимает на вход исходную строку, преобразовывает как указано в задании и возвращает новую строку.

int main() - основная функция, в которой выделяется память под исходную строку, после чего происходит считывание. В ней же происходит вызов двух вышеперечисленных функций и вывод выходной строки в консоль и файл *result.txt*.

Исходный код и листинг программы представлены в приложении А.

Примеры работы программы.

Таблица 2 — Примеры работы программы.

Входная строка	Выходная строка
ExAmPlE nUmBeR 1	example number 8
EXAMPLE NUMBER 2	example number 7
example №0123	example №9876
0123456789	9876543210

Выводы.

Было изучено представление и обработка символьной информации — строки символов на языке Ассемблер.

Была написана программа, обрабатывающая строку по определенному принципу на языке высокого уровня (C++) с включением фрагмента на языке Ассемблер по принципу встраивания (in-line).

ПРИЛОЖЕНИЕ А
ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: lb4.cpp

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

#define N 81
using namespace std;

void printINFO()
{
    cout << "------------------------------------------------------------" << '\
n';
    cout << "| Программу разработала: Чебесова Ирина, группа 9383        |" << '\
n';
    cout << "| Задание: преобразовать все заглавные латинские буквы      |" << '\
n';
      cout << "| входной строки в строчные, а десятичные цифры в           |" <<
'\n';
      cout << "| инверсные, остальные символы входной строки передаются в  |" <<
'\n';
      cout << "| выходную строку непосредственно                           |" <<
'\n';
    cout << "------------------------------------------------------------" << '\
n';
    cout << '\n';
}

char* change(char* str_in)
{
    char* str_out = new char[N];
    asm(
        "mov r8, %0\n" //записываем в регистр адрес начала выходной строки
        "mov rdi, %1\n" //записываем в регистр адрес начала входной строки


        "changeChar:\n"
        "mov al, [rdi]\n" //берем текущий символ
        "inc rdi\n" //сдвигаемся к следующему символу
        "cmp al, 0\n" //проверяем является ли символ концом строки
        "je end\n"  //если это так, то прыгаем к концу


        //меняем заглавные на прописные
        "cmp al, 0x5a\n"  //сравниваем с Z
            "jg writeChar\n" //если символ имеет больший код, а следовательно идет
дальше в таблице, то сразу переходим к выводу
            "cmp al, 0x41\n" //сравниваем сивол с A
            "jl changeNumber\n" //если он имеет меньший код - то это цифра -
уходим в проверку цифры
            "add al, 0x20\n"  //добавлем к коду символа 20, тем самым смещая его в
таблице на строчную букву (A - имеет код 0x41, a - имеет код 0x61)
            "jmp writeChar\n" //записываем символ в выходную строку


        //инвертируем цифры
```

```
        "changeNumber:\n"
            "cmp al, 0x30\n"  //сравниваем текущий символ с цифрой 0
            "jl writeChar\n"  //если меньше, чем цифра 0 - то сразу пишем
            "cmp al, 0x39\n"  //сравниваем текущий символ с цифрой 9
            "jg writeChar\n"  //если больше, чем цифра 9 - то сразу пишем
            "sub al, 0x30\n"  //вычитаем, чтобы получить цифру - например 0x33 (3)
- 0x30 (0) = 3
        "sub al, 9\n"     //вычитаем 9, чтобы получить обратную десятичную цифру
со знаком "-" 3-9=-6
        "neg al\n"        //убираем знак "-"
            "add al, 0x30\n"  //добавляем обратно для получения цифры


        "writeChar:\n"
        "mov [r8], al\n" //записываем в выходную строку символ
        "inc r8\n" //сдвигаем на следующий символ выходную строку, в которую будем
после записываеть
        "jmp changeChar\n"  //снова идем в первоначальную функцию - цикл


        "end:\n"
        :"=m"(str_out) //выходная строка с номером 0
        :"m"(str_in)  //входная строка с номером 1
      );
    return str_out;
}

int main(){
    printINFO();

    char str_in[N];
    fgets(str_in, N, stdin);
    char* str_out = change(str_in);

    ofstream f("result.txt");
    f << str_out;

    cout << str_out << '\n';
    return 0;
}
```