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 гр. 0383 _____ Козлов Т.В.

Преподаватель _____ Ефремов М.А.

Санкт-Петербург
2021

Цель работы.

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)

Замечания: 1) При выполнении преобразования обязательно использовать команды работы со строками; 2) При выполнении преобразования нельзя портить входную строку. Результат преобразования должен записываться в выходную строку.

Ход работы.

Вариант 4:

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

В ходе выполнения лабораторной работы была написана программа на C++ в VS. В качестве строк использовались массивы типа *char,* функция для считывания строки – *fgets().* Код на языке Ассемблер был вставлен с помощью ASM-блока *__asm* – данное ключевое слово вызывает встроенный ассемблер.

В данном блоке происходит обработка строки по заданным условиям варианта 4. Обработка происходит в цикле (основанном на безусловных переходах команды *jmp* а так же считывание очередного байта командой *lodsb* – которая копирует очередной байт из регистра ESI(т.к. используется 32-разрядный режим адресации).) Обработка данных основана на изменении кодов символов путем обычной арифметики. Для сравнения используются команда

cmp, а для безусловных переходов - *jl* и *jb*. (Подробности о сравнении даны в комментариях в программе).

Табл.1: Тестирование работы lab4

Входная строка	Результирующая строка	Комментарий
TimCoolProger(ha)	timcoolproger(ha)	Верно
12345678910	65432108967	Верно
Super-Puper-Test_number_4	super-puper-test_number_3	Верно
The walls have ears 3301	the walls have ears 4476	Верно

Код программы см. в приложении А.

Выводы.

В ходе выполнения работы была изучена обработка строк на языке ассемблер, а так же способа использования блока кода на языке ассемблер в коде языка высокого уровня C++.

ПРИЛОЖЕНИЕ А

Lab4.cpp:

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

char instr[81];
char outstr[161];

int main()
{
   std::cout << "Hi, I'm Timofey Kozlov - the author of this program" << std::endl;
   std::cout << "Processing by 4th variant" << std::endl;
   std::cout << "Please, enter a string" << std::endl;
   fgets(instr, 81, stdin);
   instr[strlen(instr) - 1] = '\0';
   __asm
   {
      push ds
      pop es
      mov esi, offset instr
      mov edi, offset outstr
      while:
      lodsb; в al очередной символ

         cmp al, 'Z'; проверка на то, что 'A' <= al <= 'Z'
         jg skip1
         cmp al, 'A'
         jl skip1
         add al, 32
         stosb
```

```
        jmp final
        skip1:

        cmp al, '7'; роверка на то, что '0' <= al <= '7'
        jg skip2
        cmp al, '0'
        jl skip2
        mov cl, '7'
        sub cl, al
        mov al, '0'
        add al, cl
        stosb
        jmp final
        skip2:

        stosb; Если очередной символ не удовлетворяет усл. редактирования - записываем байт в строку

        final:
        mov  ecx, '\0'
        cmp  ecx, [esi] ; проверка на конец строки
        je   whileExit; Если достигнут конец строки - выходим из цикла
        jmp  while
        whileExit :
    };
    std::cout << outstr;
    std::ofstream out("output.txt");
    out << outstr;
    return 0;
}
```