

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.
Вариант 19

Студентка гр. 1383

Седова Э.А.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2022

Цель работы.

Разработка программы для обработки символьной информации на языке Ассемблера.

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).

Задание(вариант 19): Заменить введенные во входной строке латинские буквы на десятичные числа, соответствующие их номеру по алфавиту, остальные символы входной строки передать в выходную строку непосредственно.

Выполнение работы.

В начале программы подключаются необходимые библиотеки, выделяется память для входного и выходного массивов. Далее происходит считывание входной строки из консоли при помощи библиотечной функции `getline()`. Управление переходит в ассемблерную вставку.

Описание меток:

1)start: с помощью команды `lodsb` загружаем в регистр `al` символ входной строки `input`, сравниваем с `'A'`, `'Z'`, `'a'` и `'z'` и переходим к нужным меткам.

2)letter_lower: меняет ascii-код строчной буквы на код соответствующей заглавной буквы.

3)letter_A_I: заменяет букву в диапазоне от А до I на её индекс в алфавите. Если код буквы больше либо равен коду буквы 'J' то с помощью специального ключевого слова jge, переходим к метке letter_J_S. Иначе, заполнив регистр al переходим к метке save_sumbol.

4)letter_J_S: заменяет букву в диапазоне от J до S на её индекс в алфавите. Если код буквы больше либо равен коду буквы 'T' то с помощью специального ключевого слова jge, переходим к метке letter_T_Z. Иначе, заполнив регистры al и al переходим к метке save_letter.

5)letter_T_Z: заменяет букву в диапазоне от T до Z на её индекс в алфавите. Заполнив регистры al и al переходим к метке save_letter.

6)save_sumbol: сохраняем регистр al в output и снова переходим к метке start.

7) save_letter: сохраняем регистр ax в output, далее переходим к метке next.

8)next: сравниваем текущий символ с '\0', если они не равны то вновь переходим к метке start, иначе ассемблерная вставка заканчивается.

По завершении программы выходная строка выводится в консоль и записывается в файл 'out.txt'.

Таблица 1 – примеры тестовых случаев

| Номер | Входные данные | Выходные данные |
|-------|----------------------------|--------------------------------|
| 1 | аБвГиЙ_:89+AaBaHhTtJjZz | аБвГиЙ_:89+112188202010102626 |
| 2 | abcdefWXYZ()&3940584905840 | 12345623242526()&3940584905840 |
| 3 | 678997858679&*&^&^&(_(*(&* | 678997858679&*&^&^&(_(*(&* |
| 4 | Veбра_349434_*(^*)_Бебра | 252181_349434_*(^*)_Бебра |

Программный код см. в приложении А.

Вывод.

В ходе выполнения работы была разработана программа для обработки символьной информации на языке Ассемблера.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: 4lb_Sedova_Ella.cpp

```
#include <iostream>
#include <fstream>
#include <cstdio>
#include <windows.h>

char input[81];
char output[162];
short int count = 0;

int main() {
    SetConsoleCP(1251);
    SetConsoleOutputCP(1251);
    std::cout << "Седова Элла 1383.\nВариант 19.Заменить введенные во
входной строке латинские буквы на десятичные числа,\nсоответствующие их
номеру по алфавиту, остальные символы входной строки\nпередать ввыходную
строку непосредственно.\n";
    std::cout << "Введите строку: ";
    std::cin.getline(input, 81);
    std::ofstream file("out.txt");
    __asm {
        push ds
        pop es
        mov esi, offset input
        mov edi, offset output
    start :
        lodsb
        cmp al, 65      //al<'A'
        jl save_sumbol
        cmp al, 90      //al<='Z'
        jle letter_A_I
        cmp al, 97      //al<'a'
        jl save_sumbol
        cmp al, 122     //al<='z'
        jle letter_lower
    letter_lower:
        and al, 1011111b //делает строчную букву заглавной
    letter_A_I :
        cmp al, 74      //'J'
        jge letter_J_S  //al>='J'
        sub al, 65      //'A'
        add al, 31h
        jmp save_sumbol
    letter_J_S :
        cmp al, 84      //'T'
        jge letter_T_Z  //al>='T'
        mov ah, al
        sub ah, 74      //'J'
        add ah, 30h
        mov al, '1'
        jmp save_letter
    }
```

```

letter_T_Z :
    mov ah, al
    sub ah, 84      //'T'
    add ah, 30h
    mov al, '2'
    jmp save_letter
save_letter :
    stosw
    jmp start
save_sumbol:
    stosb
next :
    cmp[esi], '\0'
    jne start
}
std::cout << "Error: " << output;
file << output;
file.close();
return 0;
}

```