

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

Студентка гр. 0383

Арсентьева. Д.А.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2021

Цель работы.

Изучение представления и обработки символьной информации с использованием строковых команд.

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).

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.

Выполнение работы.

Вариант 21: Заменить введенные во входной строке латинские буквы на числа, соответствующие их номеру по алфавиту, представленному в шестнадцатиричной СС, остальные символы входной строки передать в выходную строку непосредственно.

Была создана программа на языке C++, ассемблерную часть которой включена в программу по принципу встраивания (in-line). Использовалась кодировка cp866 (для кириллицы). Программа считывает строку и в цикле, загружая каждый символ в регистр AL командой lodsb, определяет, необходимо ли выполнить преобразования в соответствии с заданием (в коде есть комментарии). Если очередной символ не удовлетворяет усл. редактирования - записываем байт в выходную строку. Иначе он будет изменен и также записан. В конце происходит вывод результирующей строки символов на экран и ее запись в файл (output.txt).

Тексты исходных файлов программ см. в приложении А.

Тестирование.

Результаты тестирования представлены в табл. 1.

Таблица 1 – Результаты тестирования

№ п/п	Входные данные	Выходные данные	Комментарии
1.	AaBbUuKkQq*!ДдГг19	00111414Aa1010*!ДдГг19	Все верно
2.	Вдиягн148496*№%:()	Вдиягн148496*№%:()	Все верно
3.	Aa!Bb!i!Jj!Kk!Ll	00!11!88!99!Aa!Bb	Все верно
4.	Oo!Pp!Qq!Rr!Yy!Zz	Ee!Ff!1010!1111!1818!1919	Все верно

Выводы.

Были изучены представление и обработка символьной информации с использованием строковых команд.

ПРИЛОЖЕНИЕ А

ТЕКСТЫ ИСХОДНЫХ ФАЙЛОВ ПРОГРАММ

Название файла: Lab4.cpp

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

char in_str[81];
char out_str[161];

//0 - 48, 9 - 57, A - 65, Z - 90, a - 97, z - 122,

int main() {
    system("chcp 1251");
    std::cout << "The Latin letters entered in the input line will
be replaced by the \
    numbers corresponding to their alphabetical number represented in the
hexadecimal CC, \
    and the rest of the characters of the input line will be sent directly
to the output line.";
    std::cout << '\n' << "Arsentieva Daria, 0383" << '\n';
    std::cout << "Please, enter a line.." << '\n';
    std::fgets(in_str, 81, stdin);
    in_str[strlen(in_str) - 1] = '\0';
    __asm {
        push ds
        pop es
        mov esi, offset in_str
        mov edi, offset out_str
    l:
        lodsb
        cmp al, 'Z' ; проверка на то, что 'A' <= al <= 'Z'
        jg skip1
        cmp al, 'A'
        jl skip1
        mov cl, al
        sub cl, 17 ; -65+48 - убрали разницу между 0 и A
        cmp al, 'K' ; K - 75, проверка на то, что 0 <= al <= 9
        jl ret1
        cmp al, 'Q' ; Q - 81, проверка на то, что 16 <= al <=
25
        jl case3
        sub cl, 16
        jmp ret1

    skip1:
        cmp al, 'z' ; проверка на то, что 'a' <= al <= 'z'
        jg skip2
        cmp al, 'a'
        jl skip2
        mov cl, al
        sub cl, 49 ; -97+48 - убрали разницу между 0 и а
```

25

```

cmp al, 'k' ; k - 107, проверка на то, что 0 <= al <= 9
j1 ret1
cmp al, 'q' ; q - 113, проверка на то, что 16 <= al <=

```

```

j1 case3
sub cl, 16
jmp ret1

```

```

case3:
sub cl, 10 ; 10 <= al <= 15
mov al, 49 ; '1' - 49
stosb
jmp ret1

```

```

ret1:
mov 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 lExit ; Если достигнут конец строки - выходим

```

из цикла

```

jmp l
lExit :

```

```

};
std::cout << out_str;
FILE* f;
fopen_s(&f, "output.txt", "w");
if (f) {
    fwrite(out_str, sizeof(char), strlen(out_str), f);
    fclose(f);
} return 0;

```

}