

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

Петровская Е.С.

Преподаватель

Ефремов М.А.

Санкт-Петербург

2021

Цель работы.

Изучение представления и обработки символьных данных помощью 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).

Вариант 1: Формирование выходной строки только из цифр и русских букв входной строки.

Выполнение работы

В процессе выполнения данной лабораторной работы была создана программа на языке C++ с использованием встроенного ассемблера. Сама программа была написана с использованием Visual Studio. Для упрощения работы с символами русского языка была использована кодировка cp1251.

Программа принимает на вход строку длиной 80 символов и, в последствии, ее обработка происходит в блоке `__asm`, где посимвольно проверяется ее содержимое и, если встречаемый символ является символом кириллицы или числом, он записывается в результирующую строку — иначе символ пропускается.

Выходная строка выводится на экран и записывается в текстовый файл.

Разработанный код см. в Приложении А

Тестирование

№	Входные данные	Выходные данные	Комментарий
1	ыльч12hdj	ыльч12	Верно
2	kx1kмапаЛОГ	1апаЛОГ	Верно
3	1т23т12356g.	1т23т12356	Верно

Выводы.

В ходе выполнения лабораторной работы были изучены представление символьных данных и принципы работы строковых команд, используемых для их обработки.

ПРИЛОЖЕНИЕ А

ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: source.asm

```
/*Write a prog to process char info, following function realization
is a must:
* - initialization in C (or other lang): title table output with
kind of transformation and *autor of the prog
* - Char string input size <= Nmax(=80) from keyboard to the chosen
memory location(in C); *if string size > Nmax - other chars are to be
ignored
* - Transformation of the original string and write the result in
the output string in ASM
* - Output of the result char string to the screen and write in
file (in C)
* put the ASM part of prog using in-line
* DONT touch input string or change it
* when transforming using string commands is a must
* Transform VAR 1: Form result string from numbers and russian
letters ONLY*/

/**PROG:**/

#include <iostream>
#include <fstream>

char s[81];
char outstr[81];

int main() {
    system("chcp 1251"); //so the russian chars are available
                        //by default CMD has cp866 encoding
while .txt files have cp1251 encoding (at least in Win)
                        //thus it's easier to change encoding in
CMD so .txt outfile doesn't get gibberish data

    std::cout << "Var 1: Form result string from numbers and
russian letters ONLY" << std::endl;
    std::cout << "Petrovskaya Evgeniya, 0383" << std::endl;

    fgets(s, 81, stdin);
    s[strlen(s) - 1] = '\0';

    __asm {
        PUSH DS
        POP ES                ; ES = DS, stack empty
        MOV ESI, OFFSET s     ; put start of the s string
into ESI
        MOV EDI, OFFSET outstr ; put start of the outsrt into
EDI

        MOV ECX, '\0'
        Lp :                  ; loop start
```

```

        CMP ECX, [ESI]          ; check if EOL is reached
        JE stop                ; if so - exit

        LODSB                  ; load byte at address DS:SI
into AL, in x86 uses ESI
        ; check for 0 - 9
        CMP AL, '0'            ; check if AL >= '0' if not -
jmp to next check
        JB Acheck
        CMP AL, '9'            ; check if AL <= '9' if not -
jmp to next check
        JA Acheck
        JMP write              ; if '0' <= AL <= '9' - write
it into outsting

        ; check for A - я      Also let`s mark
that in cp1251 letters A-я have 192-255 numbers unlike in cp866
        Acheck :
        CMP AL, 192            ; check if AL >= 'A' if not -
jmp to next loop iteraton
        JB Lp
        CMP AL, 255            ; check if AL <= 'я' if not -
jmp to next loop iteraton
        JA Lp
        JMP write              ; if 'A' <= AL <= 'я' - write
it into outsting

        write :
        STOSB                  ; by default save AL into
address DS:DI, in x86 uses EDI

        JMP Lp                  ; next loop iteration

        stop :
};

std::cout << outstr;
std::ofstream out("output.txt");
out << outstr;

return 0;
}

```