

**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**

Студент гр. 9383

\_\_\_\_\_

Гордон Д.А.

Преподаватель

\_\_\_\_\_

Ефремов М.А.

Санкт-Петербург

2020

### **Цель работы.**

Разработать программу для обработки строки на ЯВУ со inline-вставкой на языке ассемблер.

### **Текст задания.**

Разработать программу обработки символьной информации, реализующую функции:

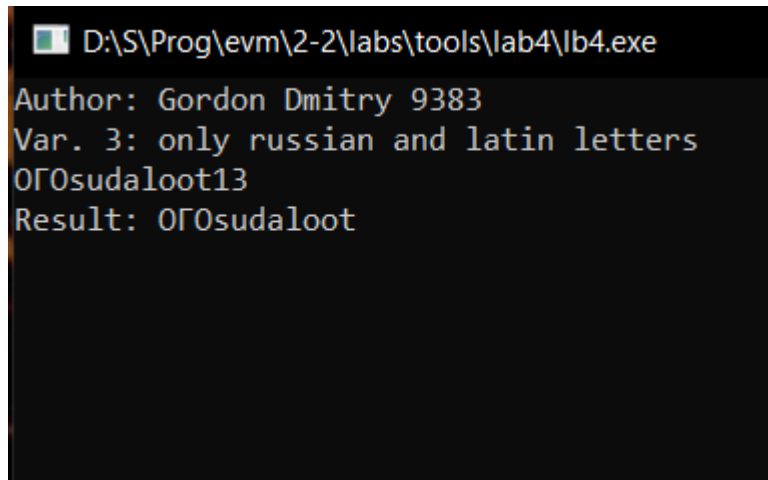
- инициализация (вывод титульной таблички с указанием вида преобразования и автора программы) - на ЯВУ;
- ввода строки символов, длиной не более  $N_{\max}$  ( $\leq 80$ ), с клавиатуры в заданную область памяти - на ЯВУ; если длина строки превышает  $N_{\max}$ , остальные символы следует игнорировать;
- 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 №3**

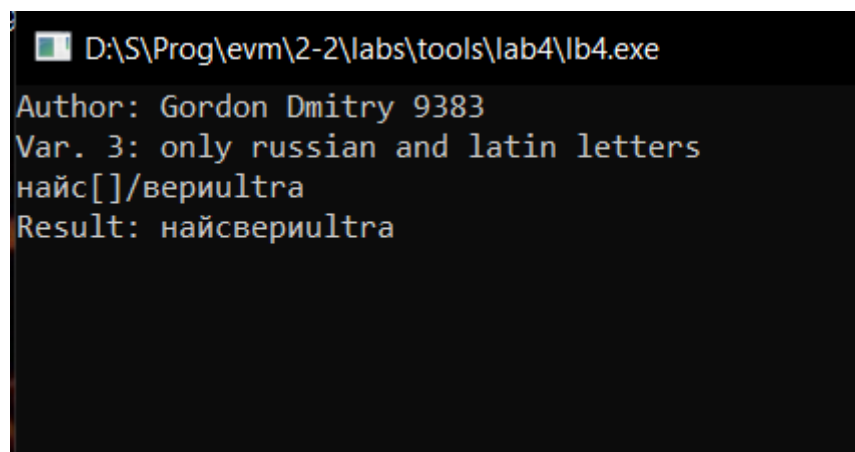
Формирование выходной строки только из русских и латинских букв входной строки.

## ПРОТОКОЛ



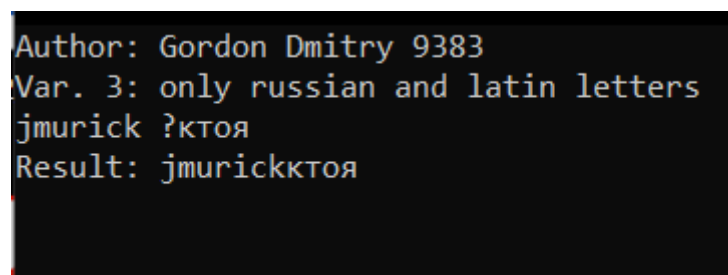
```
D:\S\Prog\evm\2-2\labs\tools\lab4\lb4.exe
Author: Gordon Dmitry 9383
Var. 3: only russian and latin letters
ОГОsudaloot13
Result: ОГОsudaloot
```

Рисунок 1 – Пример работы программы



```
D:\S\Prog\evm\2-2\labs\tools\lab4\lb4.exe
Author: Gordon Dmitry 9383
Var. 3: only russian and latin letters
найс[]/вериultra
Result: найсвериultra
```

Рисунок 2 – Пример работы программы



```
Author: Gordon Dmitry 9383
Var. 3: only russian and latin letters
jmurick ?ктя
Result: jmurickктя
```

Рисунок 3 – Пример работы программы

```
Author: Gordon Dmitry 9383  
Var. 3: only russian and latin letters  
ZzЯя  
Result: ZzЯя
```

Рисунок 4 – Пример работы программы

## **ВЫВОДЫ**

Поставленная задача была выполнена – реализована программа на ЯВУ со вставкой на языке ассемблер. Программа с помощью вставки обрабатывала строку, записывая в новую строку только русские и латинские символы. В ходе работы были использованы условные переходы.

## ПРИЛОЖЕНИЕ КОД ПРОГРАММЫ

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

char* foo(char* src)
{
    char* newstr = new char[80 + 1];
    newstr[0] = '\0';
    asm(".intel_syntax noprefix\n\t"
        "mov rsi, %1\n"           // адрес начала src
        "mov rdi, %0\n"           // адрес начала newstr

        "char_loop:\n"           // идем по всем символам src
        "mov al, [rsi]\n"         // записываем символ в al

        "cmp al, 0\n"             // сравниваем с 0 терминалом
        "je exit\n"               // если это он, то выходим

        "cmp al, 122\n"
        "jb latin\n"
        "cmp al, 239\n"
        "jb russian\n"
        "inc rsi\n"
        "jmp char_loop\n"

        "latin:\n"
        "cmp al, 122\n"
        "jg russian\n"           // больше 'z' => может русский символ
        "cmp al, 97\n"
        "jb check_latin\n"       // меньше 97 => проверяем нет ли та
м в промежутке от 90 до 97
        "jmp write_char\n"       // остальное подходит
        "check_latin:\n"         // al > 65 и < 97
        "cmp al, 90\n"           // al > 90 и < 97
        "jg char_loop\n"         // это не подходит, значит идем к с
лед символу
        "jmp write_char\n"
        "russian:\n"
        "cmp al, 128\n"
```

```

        "jb char_loop\n"    // если код символа находится за бу
квой А
        "cmp al, 239\n"
        "jg char_loop\n"    // если код символа находится после
буквы я

        // все остальное подходит

        "write_char:\n"
        "mov [rdi], al\n"
        "inc rdi\n"
        "inc rsi\n"
        "jmp char_loop\n"

        "exit:\n"
        : "=m"(newstr)
        : "m"(src)
    );
    return newstr;
}

int main()
{
    std::cout << "Author: Gordon Dmitry 9383\n";
    std::cout << "Var. 3: only russian and latin letters\n";
    char source[80 + 1] = {'\0'};
    //write(source);
    std::cin.getline(source, 80 + 1);
    char* res = foo(source);
    std::cout << '\n';
    std::cout << "Result: " << res;
    std::ofstream out;
    out.open("res.txt");
    out << res;
    out.close();
    delete[] res;
    int a;
    std::cin >> a;
    return 0;
}

```