

**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  
Преподаватель

\_\_\_\_\_  
\_\_\_\_\_

Самара Р.Д.  
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  $N_{\max}$  ( $\leq 80$ ), 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 13:**

Формирование номера введенной русской буквы по алфавиту и номера позиции его первого вхождения во входной строке и выдача их на экран.

## **Выполнение работы.**

Программа реализована на языке C++ со вставкой на языке ассемблера.

В функции *main()* выводится титульная табличка с указанием вида преобразования и автора программы. Затем считывается строка, которая обрабатывается на языке ассемблера. При помощи команды *lods b* загружается символ в регистр AL. Далее циклически программа ищет повторение этого символа, сохраняет первое вхождение, сохраняет номер этого символа в русском алфавите. Отдельно проверяются буквы ё и Ё. Если буква находится по алфавиту после Е (или е), то происходит дополнительное смещение, так как буквы Ё и ё не находятся в упорядоченном алфавите. Результат программы выводится в консоль и записывается в файл *output.txt*.

Результаты тестирования представлены в табл. 1.

Текст исходного файла программы см. в приложении А.

Таблица 1. Проверка работы программы.

№	Входная строка	Выходная строка	Комментарий
1	ЁёдбвЩгдЫ1234 ABCD E fghij Щёбв .;'.	Символ строки: 'Ё', номер первого вхождения: 1, номер буквы в алфавите: 8. Символ строки: 'д', номер первого вхождения: 3, номер буквы в алфавите: 5. Символ строки: 'б', номер первого вхождения: 4, номер буквы в алфавите: 2. Символ строки: 'в', номер первого вхождения: 5, номер буквы в алфавите: 3. Символ строки: 'Щ', номер первого вхождения: 6, номер буквы в алфавите: 27. Символ строки: 'г', номер первого вхождения: 7, номер буквы в алфавите: 4. Символ строки: 'Ы', номер первого вхождения: 9, номер буквы в алфавите: 29.	Верно

### Выводы.

В ходе выполнения данной лабораторной работы была изучена работа со строками на языке ассемблера, а также использована возможность “inline” встраивания asm кода в C++.

# ПРИЛОЖЕНИЕ А

## ТЕКСТЫ ИСХОДНЫХ ФАЙЛОВ ПРОГРАММ

Название файла: **lb4.cpp**

```
#include<Windows.h>
#include<cstdio>
#include<cstdlib>

//Формирование номера введенной русской буквы по алфавиту и номера позиции его первого вхождения
во входной строке и выдача их на экран
int main() {
    SetConsoleCP(1251);
    SetConsoleOutputCP(1251);

    const int N = 80;
    char c;
    char* str = (char*)calloc(N, sizeof(char));
    int i = 0;

    printf("Автор: Самара Р.Д. \nЗадание: Формирование номера введенной русской буквы по
алфавиту и номера позиции его первого вхождения во входной строке и выдача их на экран\n");
    printf("Введите строку: ");

    while ((c = getchar()) != '\n') {
        //if ((c >= -64 && c <= -1) || c == -72 || c == -88)    //-72 == ё, -88 ==
Ё
        str[i++] = c;
    }
    char* str1 = (char*)calloc(i * 3, sizeof(char));

    __asm {
        mov esi, str // в si (индекс источника) пометим адрес str
        mov edi, str1 // di (индекс приемника)
        mov ecx, i // ecx - счётчик массива, i - длина строки
        for:
            lodsb // кладём байт из si в al
            stosb // из al в di
            mov c, 1 // счётчик первого вхождения встроку
            mov edx, esi // адрес строки
            mov bl, al // сохраняем текущий символ

            cmp bl, 'ё'
            je russian
            cmp bl, 'Ё'
            je russian
            cmp bl, 'А'
            jl next
            jge russian
            cmp bl, 'я'
            jg next
            jle russian
        next:
            loop for
        russian:
            mov esi, str //начало поиска с самого начала исходной строки
            lodsb // считывание символа из str
        while:
            cmp al, bl //сравнение символов
```

```

        je equal // равны
        lodsb // считывание символа
        inc c // c += 1
        jmp while
equal:
        mov al, c // запись номера первого вхождения
        stosb // пересылаем номер первого вхождения в di
        mov esi, edx // пересылаем начало строки в si
        mov al, bl //берём текущий символ из bx

        cmp al, 'ё' // проверка на ё
        jne ex1 // если != - переход на ex1
        add al, 46 // смещение для ё
ex1:
        cmp al, 'Ё' // на Ё
        jne ex2
        add al, 30
ex2:
        add al, 1 // добавочное смещение на 1
        cmp al, 'a' // сравнение с буквой а
        jge little // переход к прописным буквам
        cmp al, 'E' //проверка на буквы после E (т.к Ё и ё в диапазоне от -64 до 0
отсутствуют)
        jbe ex3
        add al, 1 // доп. смещение для букв после E
ex3:
        sub al, 'A' //смещение для заглавной
        jmp end
little :
        cmp al, 'e' //проверка на буквы после e
        jbe lab2
        add al, 1 // допсмещение для букв после e
lab2 :
        sub al, 'a' //смещение для прописной
end:
        stosb // загрузка числа в строку
        loop for //cx != 0

```

```

}
int n = i;
FILE* fout;
fopen_s(&fout, "output.txt", "w");
int nmbr;
int already[33] = {0};
for (int i = 0; i < 3 * n; i += 3) {
    if (((str1[i] < 0) && (str1[i] > -65) || str1[i] == -72 || str1[i] == -
88)) {
        if (str1[i] <= -33) { //заглавные
            if(str1[i] == -72) //ё
                nmbr = str1[i] + 80;
            else if(str1[i] == -88)
                nmbr = str1[i] + 96;
            else
                nmbr = str1[i] + 65;
        }
        else{
            nmbr = str1[i] + 33;
        }
        if (already[nmbr] == 0) {
            already[nmbr] = 1;

```

```

        fprintf(fout, "Символ строки: '%c', номер первого вхождения: %d,
номер буквы в алфавите: %d.\n", str1[i], str1[i + 1], str1[i + 2]);
        printf("Символ строки: '%c', номер первого вхождения: %d, номер
буквы в алфавите: %d.\n", str1[i], str1[i + 1], str1[i + 2]);
    }
}
fclose(fout);
return 0;
}

```

//ЁёдбвЩгдЫ1234 ABCdE fghij Щёбв .;'.

