

**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»**  
**Представление и обработка символьной информации с**  
**использованием строковых команд .**

Студент гр. 0383

\_\_\_\_\_

Девятериков И.С.

Преподаватель

\_\_\_\_\_

Ефремов М.А.

Санкт-Петербург

2021

### **Вариант № 3.**

#### **Цель работы.**

Выполнить обработку входной строки на языке ассемблера. Научиться использовать ассемблерные in-line вставки в язык высокого уровня.

#### **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  $N_{\max}$  ( $\leq 80$ ), 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 встраивания (in-line).

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

#### **Выполнение работы.**

Для выполнения этой работы была создана программа на языке C++, в которой была реализована in-line вставка на языке ассемблера, которая и выполняет обработку строки. Программа считывает символ за символом знаки из входной строки и определяет по их значению в кодировке cp866, является ли данный знак буквой английского или русского алфавита. Если знак принадлежит одному из алфавитов, то знак добавляется в выходную строку. Если нет, то пропускается. Проверка производится при помощи последовательной проверки вхождения символа в допустимые диапазоны. В итоге выходная строка выводится в терминал.

Результаты тестирования программы представлены в Табл.1.

Табл.1.

Исходные данные	Выходные данные	Проверка
Фролвпр gkhjg 687342	Фролвпр gkhjg	Верно
7789^&*kjkHНОДР	kjkHНОДР	Верно
76978ролдрjhhlJHPO	ролдрjhhlJHPO	Верно

Разработанный программный код см. **Приложение А.**

### **Выводы.**

В ходе выполнения лабораторной работы были изучены возможности C++ работы с in-line вставками ассемблера, а так же возможности обработки строковых данных средствами ассемблера.

## ПРИЛОЖЕНИЕ А

### ИСХОДНЫЙ КОД ПРОГРАММЫ

main.cpp

```
#include <iostream>
```

```
char in[81];
```

```
char out[81];
```

```
int main()
```

```
{
```

```
    setlocale(LC_ALL, "cp866");
```

```
    std::fgets(in, 81, stdin);
```

```
    in[std::strlen(in) - 1] = '\0';
```

```
    __asm {
```

```
        push ds
```

```
        pop es
```

```
        mov esi, offset in
```

```
        mov edi, offset out
```

```
l:
```

```
    lodsb
```

```
        cmp al, 32; space
```

```
        jne skip1
```

```
        stosb
```

```
        jmp final
```

```
    skip1:; A - Z
```

cmp al, 65

jb final

cmp al, 90

ja skip2

stosb

jmp final

skip2:: a - z

cmp al, 97

jb final

cmp al, 122

ja skip3

stosb

jmp final

skip3:: À - ĩ

cmp al, 128

jb final

cmp al, 175

ja skip4

stosb

jmp final

skip4:: Ń - ž

cmp al, 224

jb final

cmp al, 241

ja final

stosb

```
ja final
```

```
final:
```

```
mov ecx, '\0'
```

```
cmp ecx, [esi]
```

```
je LExit;
```

```
jmp l
```

```
LExit :
```

```
};
```

```
printf("%s\n", out);
```

```
return 0;
```

```
}
```