

**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. 1383

\_\_\_\_\_

Малых А.А.

Преподаватель

\_\_\_\_\_

Ефремов М.А.

Санкт-Петербург

2022

### **Цель работы.**

Изучить представление и обработку символьной информации с помощью строковых команд.

### **Задание.**

#### **(Вариант №14)**

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).

Вариант 14: Исключение латинских букв и цифр, введенных во входной строке при формировании выходной строки.

### **Выполнение работы.**

Разработана программа на языке Си с использованием ассемблерных вставок, производящая формирование выходной строки на основе вводимой с исключением латинских букв и цифр. Проведена трансляция программы с различными входными данными. Результаты тестирования приведены в таблице 1.

Таблица 1 – Примеры тестовых случаев

№	Входные данные	Выходные данные
1	Привет, мир!;Hello, World!	Привет, мир!;,!
2	1234567890abcxyz	
3	--=_=_=_*	--=_=_=_*
4	АбвяABCz ()	Абвя ()
5	123 !@# qwe QWE ёйцу ЁЙЦУ	!@# ёйцу ЁЙЦУ

Исходный код программы см. в Приложении А

### **Выводы.**

Изучены представление и обработку символьной информации с помощью строковых команд.

## ПРИЛОЖЕНИЕ А

### ИСХОДНЫЙ КОД ПРОГРАММЫ

Название файла: lab4.c

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#define MAXLEN 81

int main()
{
    char *input_str = (char *)malloc((MAXLEN) * sizeof(char));
    char *res = (char *)malloc((MAXLEN) * sizeof(char));
    printf("Малых Андрей, группа 1383.\nИсключение латинских букв
        и цифр, введенных во входной строке при формировании
        выходной строки.\n");
    fgets(input_str, MAXLEN, stdin);
    input_str[strlen(input_str) - 1] = '\0';
    __asm__ volatile(
        "mov rsi, %[inp_str]\n" //
        "\tmov rdi, %[res]\n"
        "\tstr_loop:\n"
        "\tlodsb\n" // Load symbol
        "\tcmp al, '0'\n"
        "\tjle set_symbol \n"
        "\tcmp al, '9'\n"
        "\tjle check_null_terminator \n"
        "\tcmp al, 'A'\n"
        "\tjle set_symbol \n"
        "\tcmp al, 'Z'\n"
        "\tjle check_null_terminator \n"
        "\tcmp al, 'a'\n"
        "\tjle set_symbol\n"
        "\tcmp al, 'z'\n"
        "\tjle check_null_terminator\n "
        "\tset_symbol:\n" // Symbol insertion in res
```

```
        "\tstosb\n"
        "\tcheck_null_terminator:\n"
        "\tcmp al, 0\n"
        "\tjne str_loop\n"
        : [res] "=m"(res)
        : [inp_str] "m"(input_str));
printf("%s\n", res);
return 0;
}
```

# ПРИЛОЖЕНИЕ В

## ТЕКСТЫ ФАЙЛОВ ДИАГНОСТИЧЕСКИХ СООБЩЕНИЙ

Название файла: lab3.lst

#Microsoft (R) Macro Assembler Version 5.10  
10/30/22 20:05:1

Page

1-1

```

0000                                AStack SEGMENT STACK
0000 000C[                            DW 12 DUP(?)
      ????
      ]

0018                                AStack ENDS

0000                                DATA SEGMENT
0000 0003                            a DW 3
0002 0000                            b DW 0
0004 0002                            i DW 2
0006 FFFB                            k DW -5
0008 0000                            i1 DW 0
000A 0000                            i2 DW 0
000C 0000                            res DW 0
000E                                DATA ENDS

0000                                CODE SEGMENT
      ASSUME cs:CODE, ds:DATA, ss:AStack

0000                                Main PROC FAR
0000 1E                                push ds
0001 25 0000                            AND ax, 0
0004 50                                push ax
0005 B8 ---- R                        mov ax, DATA
0008 8E D8                            mov ds, ax

      ; Calculate i1
000A A1 0000 R                        mov ax, a
000D 3B 06 0002 R                      cmp ax, b

0011 7E 12                            jle @F
      ; if a > b: i1 = -(4i + 3)
0013 A1 0004 R                        mov ax, i
0016 D1 E0                            sal ax, 1
0018 D1 E0                            sal ax, 1
001A 05 0003                            add ax, 3
001D F7 D8                            neg ax
001F A3 0008 R                        mov i1, ax
0022 EB 29 90                            jmp I2_a_GREATER_b
0025                                @@:
      ; if a <= b: i1 = 6i - 10

```

```

0025 A1 0004 R      mov ax, i
0028 D1 E0          sal ax, 1
002A A3 0008 R      mov i1, ax ; i1 = 2i
002D D1 E0          sal ax, 1
002F 2D 000A        sub ax, 10
0032 01 06 0008 R   add i1, ax

```

```

; Calculating i2
; if a <= b: i2 = 9 - 3(i - 1) = -3i +

```

12

#Microsoft (R) Macro Assembler Version 5.10  
10/30/22 20:05:1

Page  
1-2

```

0036 A1 0004 R      mov ax, i
0039 A3 000A R      mov i2, ax
003C D1 E0          sal ax, 1
003E 03 06 000A R   add ax, i2
0042 F7 D8          neg ax
0044 05 000C        add ax, 12
0047 A3 000A R      mov i2, ax
004A EB 17 90        jmp RES_CALC

```

```

004D I2_a_GREATER_b:
; if a > b: i2 = - (6i + 8)
004D A1 0004 R      mov ax, i
0050 D1 E0          sal ax, 1
0052 A3 000A R      mov i2, ax
0055 D1 E0          sal ax, 1
0057 03 06 000A R   add ax, i2
005B 05 0008        add ax, 8
005E F7 D8          neg ax
0060 A3 000A R      mov i2, ax

```

```

; Calculating res

```

```

RES_CALC:
0063      cmp k, 0
0063 83 3E 0006 R 00 je ABS_CALC
0068 74 0F          ; if k != 0: res = min
                    (i1, i2)
006A 39 06 0008 R   cmp i1, ax ; Compare
i1
                    and i2
006E 7F 03          jg @f
                    ; if i1 < i2
0070 A1 0008 R      mov ax, i1
0073 @@:
0073 A3 000C R      mov res, ax
0076 EB 16 90        jmp PROG_END
0079 ABS_CALC:
                    ; if k == 0: res = |i1
                    + i2|
0079 A3 000C R      mov res, ax ; res = i2
007C A1 0008 R      mov ax, i1

```

```

007F 01 06 000C R      add res, ax
0083 83 3E 000C R 00      cmp res, 0
0088 7D 04              jge PROG_END
                        ; if res is neg
                        active
008A F7 1E 000C R      neg res
008E                      PROG_END:
008E CB              ret
008F                      Main ENDP
008F                      CODE ENDS
                        END Main

#Microsoft (R) Macro Assembler Version 5.10
10/30/22 20:05:1
ols-1
Symb

```

# Segments and Groups:

Class	N a m e	Length	Align	Combine
	ASTACK . . . . .	0018	PARA	STACK
	CODE . . . . .	008F	PARA	NONE
	DATA . . . . .	000E	PARA	NONE
Symbols:				
	N a m e	Type	Value	Attr
	A . . . . .	L WORD	0000	DATA
	ABS_CALC . . . . .	L NEAR	0079	CODE
	B . . . . .	L WORD	0002	DATA
	I . . . . .	L WORD	0004	DATA
	I1 . . . . .	L WORD	0008	DATA
	I2 . . . . .	L WORD	000A	DATA
	I2_A_GREATER_B . . . . .	L NEAR	004D	CODE
	K . . . . .	L WORD	0006	DATA
= 008F	MAIN . . . . .	F PROC	0000	CODE Length
	PROG_END . . . . .	L NEAR	008E	CODE
	RES . . . . .	L WORD	000C	DATA
	RES_CALC . . . . .	L NEAR	0063	CODE
	@0 . . . . .	L NEAR	0025	CODE
	@1 . . . . .	L NEAR	0073	CODE
	@CPU . . . . .	TEXT	0101h	
	@FILENAME . . . . .	TEXT	LAB3	
	@VERSION . . . . .	TEXT	510	



96 Source Lines  
96 Total Lines  
24 Symbols

47994 + 461297 Bytes symbol space free

0 Warning Errors  
0 Severe Errors