

Министерство образования Республики Беларусь
Учреждение образования
«Брестский государственный технический университет»
Кафедра ИИТ

Лабораторная работа №1
за IV семестр
по дисциплине: **КСИС**
Тема: Обработка строк.

Выполнил
студент 2 курса
Корнаевич И. Д.

Проверил
Савицкий Ю. В.

Брест 2021

Задание: Написать программу обработки текста на ассемблере. Если текст начинается и заканчивается на разные цифры, то перевернуть его, иначе заменить все буквы их кодами mod 10.

text.asm

```
1 section .data
2 text byte "1Some text2", '\0'
3
4 section .text
5 global _start
6
7 _start:
8     push text
9     call strlen
10    add rsp, 8
11
12    push rax
13    push text
14    call checkCondition
15
16    cmp rax, 1
17 .first:
18 reverse text cmp [esi], '\0'
19     je .reverse
20 .second:
21
22 ;long checkCondition (char* string, long len)
23 ;1 - first condition
24 ;2 - second
25 checkCondition:
26     push rbp
27     mov rbp, rsp
28     mov rsi, [rbp + 16]
29     mov rax, [rbp + 24]
30
31     push byte [rsi] ;first symbol in text.
32
33     mov dl, [rsi + rax]
34     pop bl
35
36     mov cl, dl
37     call .isNum
38     cmp rax, 0
39     je .second
40
41     mov cl, bl
42     call .isNum
43     cmp rax, 0
44     je .second
45
46     cmp dl, bl
47     jne .second
```

```

48
49 .first:
50     mov rax, 1
51     jmp, .done
52
53 .second:
54     mov rax, 2
55
56 .done:
57     mov rsp, rbp
58     pop rbp
59     ret
60
61 ;.isNum(char cl)
62 .isNum:
63     mov rax, 1
64     cmp cl, '0'
65     jl .notNum
66     cmp cl, '9'
67     jg .notNum
68     ret
69 .notNum:
70     mov rax, 0
71     ret
72
73 ;reverseText(char* text)
74 reverseText:
75     push rbp
76     mov rbp, rsp
77
78     mov rsi, [rbp + 16]
79 .nextChar:
80     cmp [rsi], '\0'
81     je .reverce
82     lodsb
83     push al
84     jmp .nextChar
85 .reverce:
86     mov rsi, [rbp + 16]
87 .nextRevercedChar:
88
89     cmp [rsi], '\0'
90     je .done
91     pop [rsi]
92     inc rsi
93     jmp .nextRevercedChar
94
95 .done:
96     mov rsp, rbp
97     pop rbp
98     ret
99

```

```
100
101 ;replaceByCode(char* text)
102 reblaceByCode:
103     push rbp
104     mov rbp, rsp
105     mov rsi, [rbp + 16]
106 .loop:
107     lodsb
108     cmp al, '\0'
109     je .done
110
111     div byte 10
112     mov [rsi], dl
113     jmp .loop
114 .done:
115     mov rsp, rbp
116     pop rbp
117     ret
```

Вывод: Разработана программа обработки строк на ассемблере. Изучены команды **lodsb**, **stosb**, **movsb**.