



**«Московский государственный технический университет  
имени Н.Э. Баумана  
(национальный исследовательский университет)»  
(МГТУ им. Н.Э. Баумана)**

---

ФАКУЛЬТЕТ ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ  
КАФЕДРА КОМПЬЮТЕРНЫЕ СИСТЕМЫ И СЕТИ

## О Т Ч Е Т

по **Домашней работе** № 1

Дисциплина: Машинно-зависимые языки и основы компиляции

Название **Домашней работы**: \_\_\_\_\_ обработка символьной информации

---

Студент гр. ИУ6-42Б \_\_\_\_\_ Медведев АЕ  
(Подпись, дата) (И.О. Фамилия)

Преподаватель \_\_\_\_\_  
(Подпись, дата) (И.О. Фамилия)

Москва, 2021\_

## Цель работы:

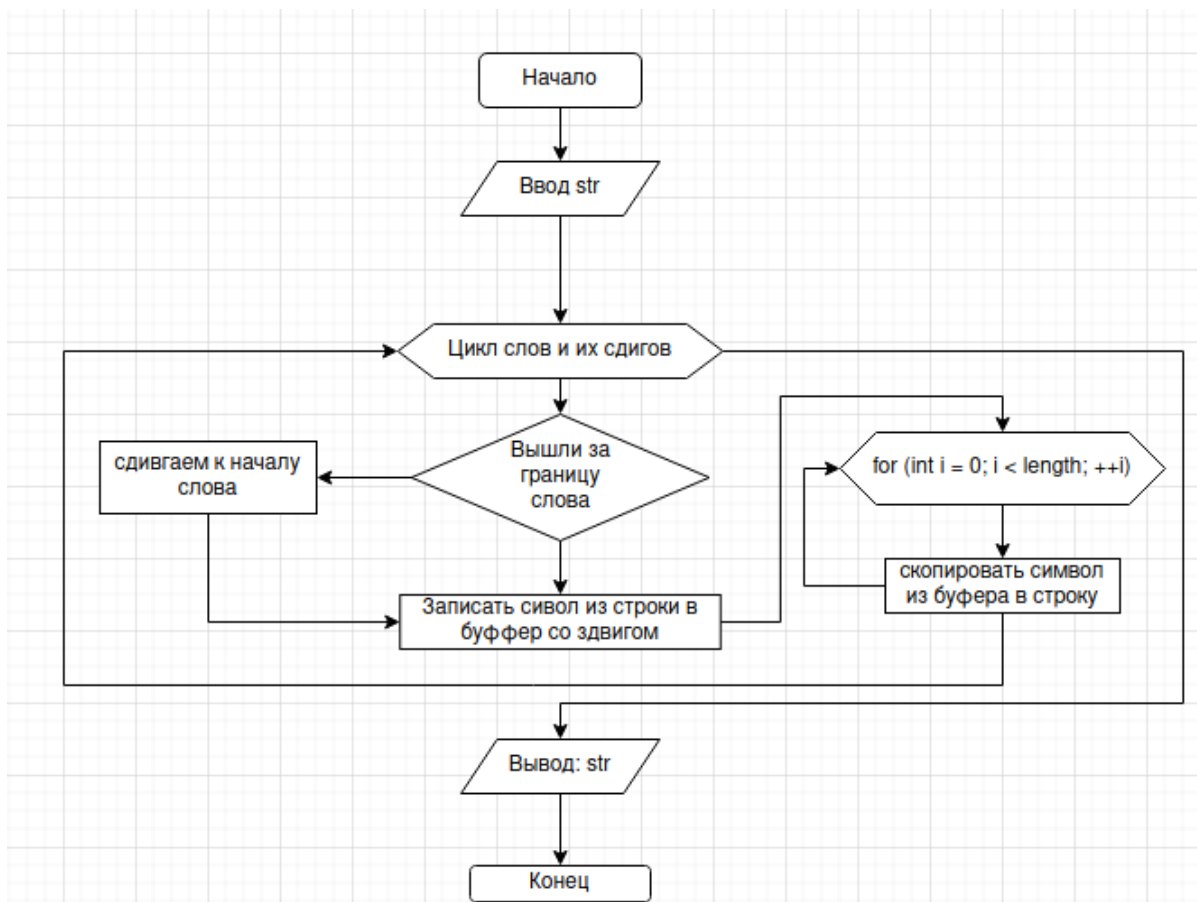
изучение команд обработки цепочек и приемов обработки символьной информации.

## Задание:

### Домашнее задание №1. Обработка символьной информации.

Дан текст 18 символов: 3 слова по 6 символов. Осуществить кольцевой сдвиг каждого слова: 1-го на 1 символ влево, 2-го на 2 символа, 3-го на 3 символа.

## Схема алгоритма:



## Код:

```
; Template for console application
.586
.MODEL flat, stdcall
OPTION CASEMAP:NONE
```

```
Include kernel32.inc
Include masm32.inc
```

```
IncludeLib kernel32.lib
```

IncludeLib masm32.lib

.CONST

MsgExit DB "Press Enter to Exit",0AH,0DH,0

.DATA

ZaprosSTR DB 13,10,'Input str (18 chars)',13,10,0

endline db 10, 13

.DATA?

buffer db 6 dup (?)

string db 18 dup (?)

inbuf DB 100 DUP (?)

.CODE

Start:

Invoke StdOut, ADDR ZaprosSTR

Invoke StdIn, ADDR string,LengthOf string

mov ecx, 3

lea edi, string

lea esi, buffer

string\_loop:

push ecx

;circle offeset on n in ax

mov eax, 4; offset

sub eax, ecx

mov ecx, 6

mov ebx, 0; i

save\_in\_buffer:

add eax, ebx

cmp eax, 6

jl sof

sub eax, 6

sof:

mov edx, [edi + eax]

mov byte ptr [esi + ebx], dl

sub eax, ebx

inc ebx

loop save\_in\_buffer

; write buffer to string

mov ecx, 6

xor eax, eax

save\_buffer\_in\_string:

mov edx, [esi + eax]

mov byte ptr [edi + eax], dl

inc eax

```

        loop save_buffer_in_string
;for string
        pop ecx
        add edi, 6
        loop string_loop

```

```

XOR     EAX,EAX
Invoke StdOut,ADDR string
Invoke StdOut,ADDR endlne
Invoke StdIn,ADDR inbuf,LengthOf inbuf

```

```

XOR     EAX,EAX
Invoke StdOut,ADDR MsgExit
Invoke StdIn,ADDR inbuf,LengthOf inbuf

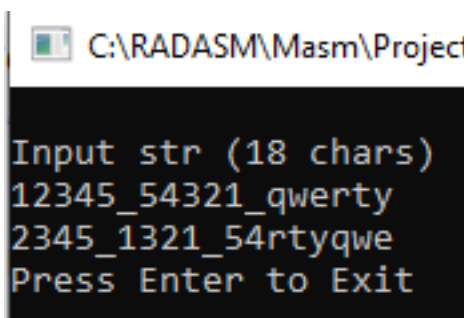
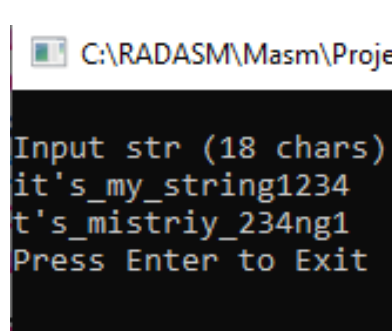
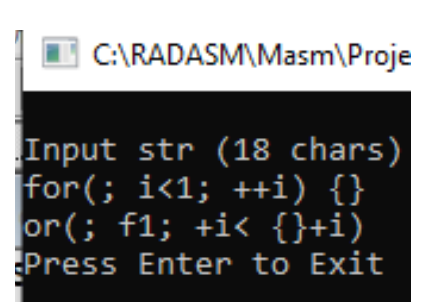
```

```

Invoke ExitProcess,0
End     Start

```

Входные данные	Ожидаемый результат	Выходные данные
12345_54321_qwerty	2345_1321_54rtyqwe	2345_1321_54rtyqwe
it`s_my_string1234	t`s_mistriy_234ng1	t`s_mistriy_234ng1
for(; i<1; ++i) {}	or(; f1; +i< {}+i)	or(; f1; +i< {}+i)

### Контрольные вопросы

1. Дайте определение символьной строки.

Строка - последовательность байт.

2. Назовите основные команды обработки цепочек?

- пересылка цепочки:

```
movs адрес_приемника,адрес_источника
```

```
movsb
```

```
movsw
```

```
movsd
```

- сравнение цепочек:

cmps адрес\_приемника,адрес\_источника  
cmpsb  
cmpsw  
cmpsd

- сканирование цепочки:

scas адрес\_приемника  
scasb  
scasw  
scasd

- загрузка элемента из цепочки:

lods адрес\_источника  
lodsb  
lodsw  
lodsd

- сохранение элемента в цепочке:

stos адрес\_приемника  
stosb  
stows  
stosd

3. Какие операции выполняют строковые команды MOVS? Какие особенности характерны для этих команд?

Эта команда выполняет примитивный перенос элемента из цепочки источника в цепочку приемник и уменьшает значение ECX на 1 и увеличивает/уменьшает значение адреса источника и приемника на 1

4. Какие операции выполняют строковые команды CMPS, SCAS? Какие особенности характерны для этих команд?

CMPS – сравнение двух цепочек, поиск несовпадающих/совпадающих элементов  
SCAS – сканирование цепочки, поиск нужного элемента

5. Как обеспечить циклическую обработку строк?

Поставить REP/REPZ/REPZ перед оператором обработки строки

6. Какова роль флага DF во флаговом регистре при выполнении команд обработки строк?

Направление обработки, 0 – вперед, 1 – назад.

7. Какие макрокоманды используются в среде RADASM для ввода и вывода строк?

- movs адрес\_приемника,адрес\_источника  
- cmps адрес\_приемника,адрес\_источника  
- scas адрес\_приемника

- lods адрес\_источника

- stos адрес\_приемника

В зависимости от типа адреса заменяются на одну из команд указанных выше

8. Как правильно выбрать тестовые данные для проверки алгоритма обработки строки?

Тесты должны быть составлены с условие возможных поломок алгоритма.

Задача тестов проверить все нетривиальные поведения и выявить неисправность.

**Вывод:** В ходе домашней работы была разработана программа на языке ассемблера, которая делает циклический сдвиг слов.