Задача:

Разработать набор процедур работы с очередью, а именно:

- включение нового элемента;

- выборка очередного элемента (со сдвигом очереди)

Разработать демо-программу

В данной работе мы учимся работать с очередью.

В задании рассмотрена работа очереди. Очередь – структура данных с дисциплиной доступа к элементам «первый пришёл – первый вышел» (FIFO, First In – First Out). Добавление элемента возможно лишь в конец очереди, выборка происходит только из начала очереди, при этом выбранный элемент удаляется.

## 1.2. Текст программы

DOSSEG

.MODEL small

.386

.STACK 100h

.DATA

fifo DB 10 dup ('-') ;очередь

fifo\_size DW 8 ;размер очереди

fifo\_end DW 0 ;указывает на конец очереди

wr\_num DW 0 ;количество записываемых элементов

rd\_num DW 0 ;количество читаемых элементов

f\_msg DB 'FIFO: ',10,13,'$'

input\_msg DB 'Enter number of input elements:',10,13,'$'

elem\_msg DB 'Enter elements of array:',10,13,'$'

output\_msg DB 'Enter number of output elements:',10,13,'$'

empty\_msg DB 'FIFO is empty!',10,13,'$'

full\_msg DB 'FIFO is full!',10,13,'$'

excess\_msg DB 'Your number become max of avilable',10,13,'$'

retry\_msg DB 'It is very big number, try again!',10,13,'$'

.CODE

;---------------------------------------

;процедура выводящая содержимое очереди

;---------------------------------------

print\_fifo proc

;вывод надписи "FIFO: "

mov ah,9

mov dx,offset f\_msg

int 21h

;вывод содержимого очереди

xor si,si

lea bx,fifo

mov cx,fifo\_size

print:

mov dl,[bx+si]

mov ah,2

int 21h

inc si

loop print

mov al,10 ;перенос строки

int 29h

int 29h

ret

print\_fifo endp

;----------------------------------------------

;процедура включения новых элементов в очередь

;----------------------------------------------

write\_fifo proc

;проверка на заполненность очереди

mov ax,fifo\_size

cmp fifo\_end,ax

je full\_label

;вывод сообщения о количестве заполняемых элементов

mov ah,9

mov dx,offset input\_msg

int 21h

;ввод количества заполняемых элементов

mov ah,0 ;ввод информации в консоль

int 16h

mov byte ptr wr\_num,al ;запись числа в переменную

sub word ptr wr\_num,30h ;преобразование ASCII кода в число

mov dl,al ;вывод введенного числа

mov ah,2

int 21h

mov al,10 ;перенос строки

int 29h

;проверка на превышение зачения wr\_num

mov bx,fifo\_end

mov ax,fifo\_size

sub ax,bx

mov bx,wr\_num

cmp bx,ax

jb excess\_label

;записываем в wr\_num разность fifo\_size и fifo\_end

mov ah,9

mov dx,offset excess\_msg

int 21h

mov bl,byte ptr fifo\_end

mov al,byte ptr fifo\_size

sub al,bl

mov byte ptr wr\_num,al

add al,30h

int 29h

mov al,10

int 29h

excess\_label:

;вывод сообщения о заполнении очереди

mov ah,9

mov dx,offset elem\_msg

int 21h

;заполнение очереди

xor si,si

add si,fifo\_end

lea bx,fifo

mov cx,wr\_num

input:

mov ah,0

int 16h

mov [bx+si],al

inc si

mov dl,al

mov ah,2

int 21h

loop input

;обновляем значение fifo\_end

mov bx,wr\_num

add fifo\_end,bx

jmp over\_full

;вывод сообщения о заполненности очереди

full\_label:

mov ah,9

mov dx,offset full\_msg

int 21h

over\_full:

mov al,10 ;перенос строки

int 29h

ret

write\_fifo endp

;--------------------------------------

;процедура чтения элементов из очереди

;--------------------------------------

read\_fifo proc

;проверка на пустоту очереди

cmp fifo\_end,0

je empty\_label

try\_again:

;вывод сообщения о количестве читаемых элементов

mov ah,9

mov dx,offset output\_msg

int 21h

;ввод количества выводимых элементов

mov ah,0 ;ввод информации в консоль

int 16h

mov byte ptr rd\_num,al ;запись числа в переменную

sub word ptr rd\_num,30h ;преобразование ASCII кода в число

;проверка на превышение зачения rd\_num

mov bx,rd\_num

cmp bx,fifo\_end

jae retry\_label

mov dl,al ;вывод введенного числа

mov ah,2

int 21h

mov al,10 ;перенос строки

int 29h

;чтение из очереди

xor si,si

lea bx,fifo

mov cx,rd\_num

output:

mov dl,[bx+si]

mov ah,2

int 21h

inc si

loop output

;сдвиг элементов

xor di,di

xor si,si

add si,rd\_num

lea bx,fifo

mov cx,fifo\_end

sub cx,rd\_num

moving:

mov al,[bx+si]

mov [bx+di],al

inc di

inc si

loop moving

mov fifo\_end,di

;зачистка

mov si,fifo\_end

mov cx,fifo\_size

sub cx,fifo\_end

erasing:

mov byte ptr [bx+si],'-'

inc si

loop erasing

jmp over\_empty

;вывод сообщения о том, что очередь пуста

empty\_label:

mov ah,9

mov dx,offset empty\_msg

int 21h

jmp over\_empty

;вывод сообщения о том, что введено большое число

retry\_label:

mov ah,9

mov dx,offset retry\_msg

int 21h

jmp try\_again

over\_empty:

mov al,10 ;перенос строки

int 29h

ret

read\_fifo endp

;--------------------------

;начало основной программы

;--------------------------

start:

mov ax,@data

mov ds,ax

xor ax,ax

call write\_fifo

call print\_fifo

call read\_fifo

call print\_fifo

call write\_fifo

call print\_fifo

call read\_fifo

call print\_fifo

mov ax,4c00h

int 21h

END start

## 1.3. Блоксхемы

### 1.3.1. Процедура записи в очередь

Очередь заполнена

Вывод сообщения о количестве заполняемых элементов

Ввод количества заполняемых элементов wr\_num

wr\_num превышает оставшееся место в очереди

Записываем в wr\_num разность fifo\_size и fifo\_end

Вывод сообщения о заполнении очереди

Заполнение очереди

Обновляем значение fifo\_end

Вывод сообщения о заполненности очереди

да

нет

нет

да

### 1.3.2. Процедура чтения из очереди

Очередь пуста

Вывод сообщения о количестве читаемых элементов

Ввод количества читаемых элементов rd\_num

rd\_num превышает количество элементов в очереди

Вывод сообщения о том, что следует ввести новое значение кв\_тгь

Чтение из очереди

Вывод сообщения о том, что очередь пуста

да

нет

нет

да

Сдвиг элементов

Зачистка

Обновляем значение fifo\_end

## 1.4. Скриншоты

Изображение выглядит как текст

Автоматически созданное описание

Изображение выглядит как текст

Автоматически созданное описание