```
; Fibonacci sequence / Only 10 elements so we can use one byte
IDEAL
MODEL         small
STACK         100h
DATASEG
seq           db 10 dup(0h)
cnt           db 0AH           ;Initialize the counter to 10 numbers.
numstr        db  '$$$$$'      ;STRING FOR 4 DIGITS.
CODESEG
;--------------------------------------------------------------------------------
proc          newline
              push ax
              push dx
              mov  ah,2
              mov  dl,10
              int  21h
              pop  dx
              pop  ax
              ret
endp          newline
;--------------------------------------------------------------------------------
proc          number2string
              push cx
              call dollars               ;prepare numstr with $ signs.
              mov  bx, 10                ;extracting a digit at a time
              mov  cx,0
;EXTRACT DIGITS
cycle1:       mov  dx,0                  ;to facilitate division
              div  bx                    ;dx:ax / 10 =  ax - quotient  dx -remainder
              push dx                    ;using a stack will allow fetching in the
correct order
              inc  cx
              cmp  ax,0                  ;done if number is zero
              jne  cycle1
;Retrieve pushed digits (opposite order)
              mov  si, offset numstr
cycle2:       pop  dx
              add  dl,48                 ;convert dec digit to ASCII - 48 is '0' in ASCII
              mov  [si],dl
              inc  si
              loop cycle2
              pop cx
              ret
endp          number2string
;--------------------------------------------------------------------------------
proc          dollars                    ;No parms. Fills area in memory with "$"
              push cx
              push di
              mov  cx, 5
              mov  di, offset numstr
loop_dollars: mov  bl, '$'
              mov  [ di ], bl
              inc  di
              loop loop_dollars
              pop  di
              pop  cx
              ret
endp          dollars
;--------------------------------------------------------------------------------
start:        mov ax, @data
              mov ds, ax
              lea si,[seq]
;print first element
              push ax
              push bx
              push si
```

```
                xor  ax,ax
                mov  al,[seq]
                call number2string      ;takes the number in al and returns ASCII in numstr
                mov  dx, offset numstr
                mov  ah,9h
                int  21h
                call newline
                pop  si
                pop  bx
                pop  ax

;end print first element
                mov cl,[cnt]     ;looping counter.
                xor ax,ax        ;Beginning numbers 0 and 1.
                mov bx,01h
                inc si
                mov [si],bx      ;array initialed to zero so start placing from the second.
;print second element
                push ax
                push bx
                push si
                xor  ax,ax
                mov  al,[si]
                call number2string      ;takes the number in al and returns ASCII in numstr
                call newline
                mov  dx, offset numstr
                mov  ah,9h
                int  21h
                call newline
                pop  si
                pop  bx
                pop  ax
;end print second element
                sub cl,2         ;First two are placed in array already before the loop.
fbloop:         add ax,bx        ;Calc fibonacci values in a loop
                inc si
                mov [si],ax
;print the rest of the elements
                push ax
                push bx
                push si
                xor  ax,ax
                mov  al,[si]
                call number2string      ;takes the number in al and returns ASCII in numstr
                mov  dx, offset numstr
                mov  ah,9h
                int  21h
                call newline
                pop  si
                pop  bx
                pop  ax
;end print rest of the elements
                mov ax,bx
                mov bx,[si]
                loop fbloop
exit:           mov ax, 4c00h
                 int 21h
END start
```