

1)

```
text    segment

        assume cs:text, ds:text

        org 100h


start:   jmp go

msg db "Sveikss!", 13, 10, "$"

go:      mov ah, 9h

        mov dx, offset msg

        int 21h

        int 20h

text    ends

        end start
```

2)

```
code    segment

        assume cs:code, ds:code

        org 100h


start:   jmp go


string  db '01234567891*ABC',0
buf     db '000000&'


go:      mov si,0
         mov ah,'*'
check:   cmp string[si],0
         je notfound
         cmp ah,string[si]
         je found
         inc si
         jmp check


found:

        ;inc ax

        mov ax,si
        mov si,5
        mov bl,10
d:       div bl
        add ah, 30h
        mov buf[si], ah
        cmp al,0
```

*je put*

*mov ah,0*

*dec si*

*jmp d*

*put:*

*mov ah,9*

*mov dx, offset buf*

*int 21h*

*jmp done*

*notfound:*

*mov dl,'?'*

*mov ah,6*

*int 21h*

*done:*

*int 20h*

*code ends*

*end start*

3)

*text segment*

*assume cs:text,ds:text*

*org 100h*

*start: jmp go*

*inbuf equ this byte*

*maxlen db 20*

*actlen db 0*

*intext db 20 dup(0)*

*chr db ?*

*msg1 db 'Ievadiet simbolu:\$'*

*msg2 db 'Ievadiet virkni:\$'*

*newline db 13,10,'\$'*

*outbuf db '000000\$'*

*go:*

*mov ah,9*

*mov dx,offset msg1*

*int 21h*

*mov ah,1*

*int 21h*

*mov chr, al*

*mov ah,9*

*mov dx, offset newline*

*int 21h*

*mov ah,9*

*mov dx, offset msg2*

*int 21h*

*mov ah, 0ah*

*mov dx, offset inbuf*

*int 21h*

*mov ah,9*

*mov dx, offset newline*

*int 21h*

*;=====*

*push ax*

*mov al, [inbuf +1]*

*mov actlen, al*

*pop ax*

*;=====*

*cmp actlen, 0*

*je notext*

*xor ax,ax*

*xor cx,cx*

*mov cl,actlen*

*xor si,si*

*mov dl,chr*

```
c:    cmp inbuf[si+2],dl
      jne next
      inc ax
next:  inc si
      loop c
```

output:

```
      mov si,5
      mov bl,10
d:    div bl
      add ah,30h
      mov outbuf[si],ah
      cmp al,0
      je put
      mov ah,0
      dec si
      jmp d
```

notext:

put:

```
      mov ah,9
      mov dx, offset outbuf
      int 21h
      int 20h
text  ends
      end start
```

4)

```
text    segment

        assume cs:text, ds:text

        org 100h


start:   jmp go


go:      ;mov dx,40h

        mov es, dx

        mov di, 40h

        mov ax, es:[di+2]


        int 20h

text    ends

        end start
```