A) Makro ve prosedurler arası fark nedir.

Prosedurler ona program olişinda yazıları, ona propram iqinde call F1 zetlinde qozrılarak belli işlemler yapılan Ponk parqaları gibidir. Propram iqine pomultu değildir. Makrolar rose ana kodunuzun bir parqası olup kod iqine gomulturler. Makroya gelince gomultu olan komut qalışır devam eder. Makrolar gerelde AX 18 X DX registerleri ile qalışır. Push, pop komutları qokca kullanılır.

1- gotoxy ch, cl reklinde makromuz matrisel işlemlerde satır ve sütun üzerinde haretet edilmesini sağlar. [push pop AX, 8X, DX]

2- putc V X Tetlinde karekterin imlecin gösterdiği hisma yazılmasını sağlar.

int 21H 816: kesmeler ile bu islemi halleder

B-) Sombol: Gelorekel programbne dollandeks degstere benser. Degstanden farki se boelongia odresni tutmasidir.

S1B) Aslanda semboller, gelenet sel programleme dillerindelei degiskenlere Karsalak ge Sembollerin Forka, Kullandan Duzmenen buslangan adresini tutmaktur 2A) Makrolando, kod makroyo gómülür. Prosedürlende ise prosedür Galisir isonra sında diğer kodlar Galismaya devam eden, Yani main prosedüre geri dönülür ve main "cail" dan i'tibaren kardığı yerden devam eder.

Makrolan diğer programlama di'llerindeki kütüphoneler gibidir.

Makrolan diğer programlama di'llerindeki kütüphoneler gibidir.

90toxy ch, cl makrosu ve sonrasında gelen putc ''
mokrosu ile ekranda belli satır, sütün değer lerine (ch, c')
putc icine koy duğumuz değeri cizdire biliyoruz.

510)

I. Yallis. ab, equ, code jibi comution directification ue directification set ionisinde bulunmoner.

2. Doğru

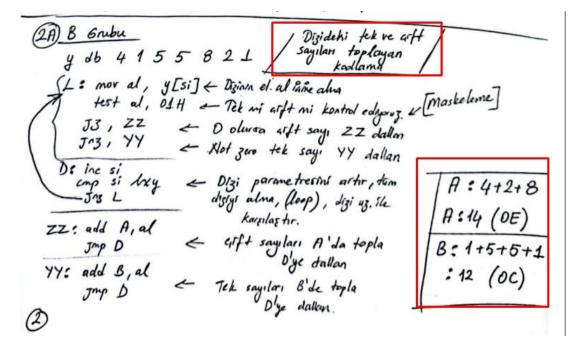
3. Long. Boiom al de kaian ah yormacinda tutulur.

4. Dogiu

5. Doğru

BX kullonlobilir.

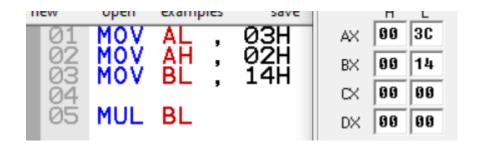
7. Janus. makrodur.



52B)

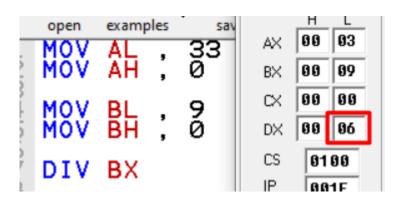
- · lx db #-x yapılmıs. (x db \$-x olmolydı.
- · mov cx, ex youlistic. Cx 16 bit, ex 8 bittir. (db olarde varimit)
- mou al, x [0-67] yanış. 57 kobul edilmer.
- · cmp si, lxy yours. Lxy nesopramanis
- e endp -> main endp olmali.
- · Fx proc rande ret exsik.

Yy ve zZ hataları hata değildir !! Büyük küçük harf aynı olarak algılanır



DX Değişmez

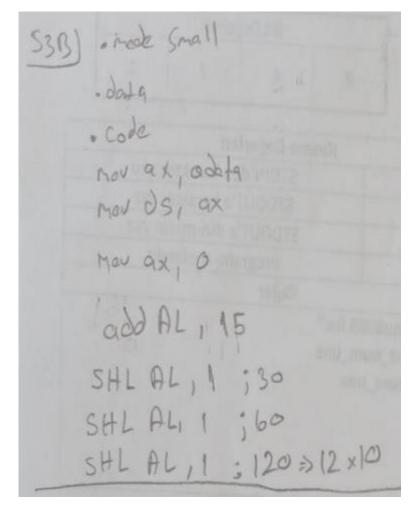
```
02 mov al , 12
03 mov bl , 12
05 shl al , 3 ; 8 ile carpma
06 shl bl , 1 ; 2 ile carpma
07
08 add al , bl
```



3 Tam -> AX 6 Kalan-> DX

```
OCH CITE
    .model small
    .data
    .code
    mov ax , @data
    mov ds , ax
    mov dl , "Y"
    mov ah , 01H
int 21H
    and al , OFH
   cmp al , 5
   mov ah
int 21H
                02H
    HLT
    BY:
         cmp al , 8 jl KC jmp D
29
30
31 KC:
32 mov
33 jmp
         dl , "D"
```

S3B İçin Farklı Yaklaşımlar



```
3B

12 × 10 = 12 × (8+2) = 12 × 23 + 12+12

mov bl, 12

mov al, 12

shl al, 3

add al, bl

add al, bl
```

```
B-) 312 = 2^{2} + 2^{3} \Rightarrow 12 \times 10 \Rightarrow 10 \cdot 4 + 10 \cdot 8

B-) madel small

code

May al, 40

Sh1 al, 2

May b1, 10

Sh1 b1, 3

add al, b1
```

```
.MODEL SMALL
    x db 10 dup(0)
mov x[si] , al
       cmp si , 10 jnz L
     cmp si , 7
jl KC
jge BY
D:
      cmp si , 0 jne L1
       mov dl , x[si]
```

```
01
02
03
04
05
                                       .MODEL SMALL
                                    .DATA
  06
07
08
09
10
                              x db 2,0,1,4,
v db 0,0,0,0
                                                                                                                                                                                                                                                       1,0,2,3,
                                                                                                                                                                                                                                                                                                                                                                                                            3,2,1,0,
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           4,2,1,5
                                      .CODE
                               mov ax , @data
mov ds , ax
   11234567
14567
                                  mov ax ,
                                  mov cx ,
                                   mov si
18 mov

19 L0:

21 223 425 67 289 01 inner                                   mov di
                                   L0: mov ch , 0
                                                                                   L1:
                                                                                mov al , x[si]
cmp al , cl
jg T
D:
                                                                                  inc si
inc ch
                                                                                  cmp ch , 4 jne L1
                              inc cl
inc di
cmp di
jne L0
                                                                                   add v[di] , al
                                                                                    jmp D
```