

Alıştırma Problemleri

Problem 1

**; Bellekte 0700:0200 adresindeki isaretsiz 8-bitlik sayılardan
; ilk 10 tanesi icerisinden en kucuk sayiyi ve sıra numarasini
; bulup bir register a kopyalayiniz**

MOV BX,0200H

MOV BYTE PTR [bx],55h
MOV BYTE PTR [bx+1],66H
MOV BYTE PTR [bx+2],54H
MOV BYTE PTR [bx+3],11H
MOV BYTE PTR [bx+4],87H
MOV BYTE PTR [bx+5],33H
MOV BYTE PTR [bx+6],75H
MOV BYTE PTR [bx+7],05H
MOV BYTE PTR [bx+8],5FH
MOV BYTE PTR [bx+9],23H

org 100h
MOV BX,0200H
MOV SI,1

MOV DL,[BX] ;en kucuk sayiyi tutan register
MOV DH,00 ;en kucuk sayiyinin dizideki sirasini tutan register

J1:

MOV AL,[BX+SI] ; 0700:0200 taban adresindeki (SI ile indekslenmiş) ilgili byte alır

CMP AL,DL
JAE devam ;ele alınan sayı daha küçük değilse

MOV DX,SI
MOV DH,DL ;yeni en küçük sayının indisi kopyalanır
MOV DL,AL ;yeni en kucuk sayi kopyalanır

devam:

INC SI

CMP SI,10
JNZ J1

HLT

Problem 2:

0, 1, 1, 2, 3, 5, 8, 13, 21, 34, ...

$$\text{Fibonacci}(n) = \begin{cases} 0 & \text{if } n = 0 \\ 1 & \text{if } n = 1 \\ \text{Fibonacci}(n-1) + \text{Fibonacci}(n-2) & \text{otherwise} \end{cases}$$

FIGURE 2-7 Fibonacci Numbers Recursive Definition

**; DL registerinde verilen sayiya en yakin iki Fibonacci sayisini bulan
; programi yaziniz
; Sonuc BH (Fibo_kucuk) ve BL(Fibo_buyuk) de tutulur**

ORG 100H

MOV DL,15

MOV AL,0 ;Fibonacci dizisinin 1.(i.) elemani

MOV BL,1 ;Fibonacci dizisinin 2.(i+1.) elemani

devam:

ADD AL,BL ; Dizinin bir sonraki elemanini hesaplar

MOV BH,BL ; (i+1). yani eski elemani yedekler

MOV BL,AL ; (i+2). elemani , yani yeni elemani (i+1). konuma yazar

MOV AL,BH ; yedeklenen eski elemani i. konuma yazar

CMP BL,DL ; yeni eleman ust limiti gecti mi?

JBE devam ; hayir ise devam

HLT ;evet ise bitir

//////////////////////////////////// Alt program kullanarak çözüm

ORG 100H

MOV CX,10

MOV SI,0200h

MOV AL,0 ;Fibonacci dizisinin 1.(i.) elemani

MOV BL,1 ;Fibonacci dizisinin 2.(i+1.) elemani

tekrar:

CALL Fibo

MOV [SI],BL

INC SI

LOOP tekrar

ret

Fibo PROC ; procedure declaration.

ADD AL,BL ; Dizinin bir sonraki elemanini hesaplar

MOV BH,BL ; (i+1). yani eski elemani yedekler

MOV BL,AL ; (i+2). elemani , yani yeni elemani (i+1). konuma yazar

MOV AL,BH ; yedeklenen eski elemani i. konuma yazar

RET ; return to caller.

Fibo ENDP

Problem 3:

**; DL registerinde ust limiti verilen degere kadar olan asal
; sayilari hesaplayip bellekte 0700:0200H adresinden
; itibaren yerlestiren programi yaziniz.**

org 100h

MOV BX,0200H ;bellek offset adresi

MOV DL,29 ; asal sayi ust limiti
mov di,00

J1:

MOV CL,DL
MOV SI,0 ; tam bolen adedini tutar

TEKRAR:

MOV AL,DL ;bolunecek sayi AX'e kopyalanir
MOV AH,0

DIV CL ;sayi kendisinden baslamayip bir azaltilarak 1'e
; kadar olan sayılara tek tek bolunur

CMP AH,0 ; kalan var mi?
JNZ kalanVar

INC SI ; kalan yok ise tam bolen adedini tutan Reg. arttırılır

kalanVar:

LOOP TEKRAR ; CX= CX-1

CMP SI,2 ; eger ilgili sayi sadece 1'e ve kendisine tam olarak bolunebiliyorsa o sayi asaldır
JNZ asalDegil

MOV [BX+DI],DL ; bulunan asal sayi bellege yazilir

INC DI ; asal sayi indeksi 1 arttırılır

asalDegil:

DEC DL ; sayiyi 1 azaltir

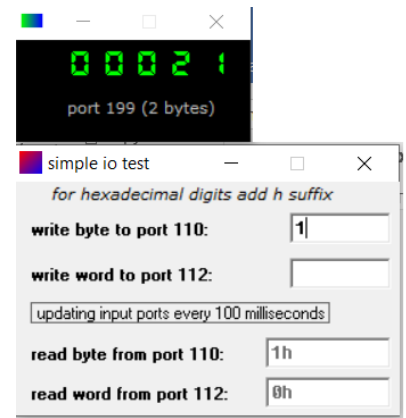
CMP DL,1 ; tum sayilar kontrol edildi mi?

JNZ J1

HLT

Problem 4:

**; EMU 8086 nin sanal portlarından klavyeden alınan artis
;miktarını bir sayacta biriktirerek sanal portlardan display
;portunda belirli bir gecikme ile yansitan program**



PORTA EQU 110 ; Klayveden giris yapmaya yarayan sanal giris portu
PORTB EQU 199 ; Sanal cikis portu, diplay

org 100h
MOV SI,1 ;displayde gosterilecek sayac degeri
MOV CX,1 ; artis miktarı

tekrar:

MOV DX,PORTA
IN AL,DX ;Giris portundan klavyeden girilen degeri oku

cbw ;ax registerindeki byte worde çevir

MOV CX,AX ;klavyeden girilen artis miktarini yedekle

ADD SI,CX ;sayactaki degeri guncelle

MOV DX,PORTB
MOV AX,SI
OUT PORTB,AX ;sayactaki degeri diplaye tasi
CALL DELAY ;displaydeki degerin gorunebilmesi icin gecikme
JMP tekrar

ret

DELAY PROC
PUSH CX ; ana programda kullanılan registerlari yigina yedekle

MOV CX,020H

J1:

NOP
NOP
LOOP J1

POP CX ;yedeklenen reg yigindan geri cek

RET

DELAY ENDP