

BLM 2021 2. ÖDEV RAPORU

Dersin Adı: Alt Seviye Programlama

Öğretim Üyesi: Dr. Öğretim Üyesi Erkan USLU

Öğrencinin Adı ve Soyadı: Seyyid İbrahim GÜLEÇ

Öğrencinin Numarası: 16011609

KAYNAK KODU :

PAGE 60,80

TITLE ornek_____

STACKSG SEGMENT PARA STACK 'STACK'

DW 512 DUP(?)

STACKSG ENDS

DATASG SEGMENT PARA 'DATA'

DIZI1 DB 0Dh,18h,1Fh,0FBh,28h,0Bh,01h,34h,14h,0FFh

DIZI2 DB 01h,02h,03h,1Bh,0Ah,0Ah,0FFh,19h,42h,0FFh

SONUC DW 10 DUP(0)

ESAYI DB 0Ah

DATASG ENDS

CODESG SEGMENT PARA 'CODE'

ASSUME CS:CODESG, DS:DATASG, SS:STACKSG

BASLA PROC FAR

PUSH DS

XOR AX,AX

PUSH AX

MOV AX,DATASG

MOV DS,AX

XOR BX,BX

MOV CX,ESAYI

L1: XOR AX,AX

MOV AL,[BX]

PUSH AX

MOV AL,[BX+10]

PUSH AX

CALL RECUR

```
SHL BX,1
POP [BX+20]
SHR BX,1
ADD BX,1
LOOP L1
RETF
```

```
BASLA ENDP
```

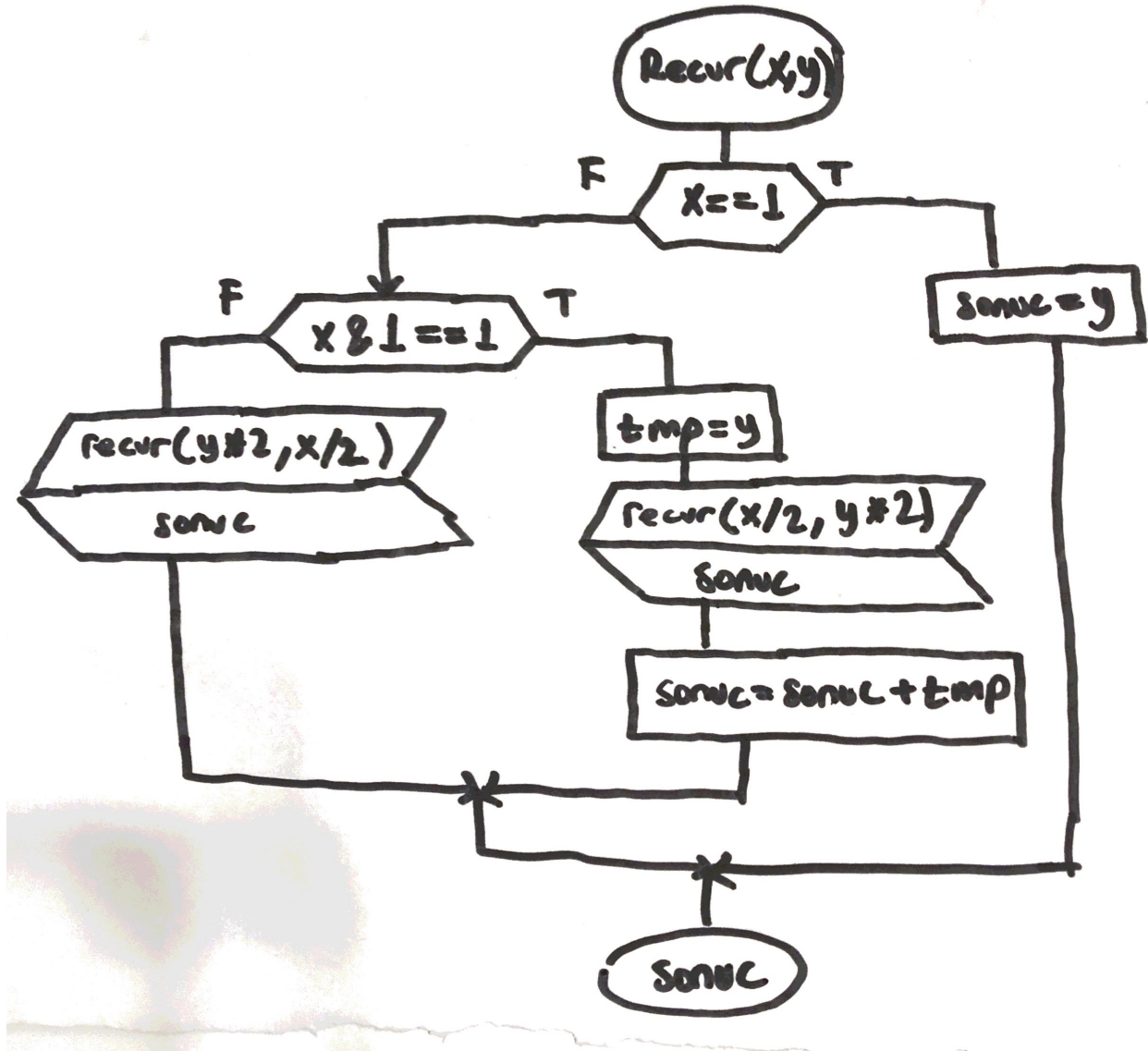
```
RECUR PROC NEAR
```

```
PUSH BP
PUSH BX
PUSH DX
MOV BP,SP
MOV BX,[BP+8]
MOV DX,[BP+10]
CMP BX,1
JNZ A1
MOV [BP+10],DX
JMP SON
```

```
A1: TEST BX,0001
JZ A2
SHR BX,1
PUSH BX
MOV BX,DX
SHL DX,1
PUSH DX
CALL RECUR
POP [BP+10]
ADD [BP+10],BX
```

```
        JMP SON
A2:     SHR BX,1
        SHL DX,1
        PUSH DX
        PUSH BX
        CALL RECUR
        POP [BP+10]
SON:    POP DX
        POP BX
        POP BP
        RET 2
RECUR   ENDP
CODESG  ENDS
END BASLA
```

a) 002C adresinden itibaren yerleştirilmiş olan özyinelemeli fonksiyonun akışını çiziniz (üst seviye tanımlarla).



—> Yapılan bölme ve çarpma işlemleri assembly kodunda shift right ve shift lefttir ve integer bölmesi şeklinde yapılır!

b) Ana prosedür BX=1 değeri ile yürütüldüğünde yığının en dolu olduğu durumda yığın adres ve içeriğini gösteriniz. (yığındaki değerlerin nereden geldiğini yazarak).

```
076A:03D0  00 00 00 00 00 00 00 00 00 00 02 00 E4 03 2D 00  ..
076A:03E0  AD 07 A3 01 30 00 01 00 F0 03 61 00 01 00 30 00  ..
076A:03F0  00 00 01 00 00 00 1B 00 02 00 18 00 00 00 5A 07  ..
076A:0400  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ..
```

```
-r
AX=0002 BX=0001 CX=0009 DX=0030 SP=03E4 BP=03E4 SI=0000 DI=0000
DS=07AA ES=075A SS=076A CS=07AD IP=002D  NU UP EI PL NZ AC PE NC
07AD:002D 8B5E08      MOV     BX,[BP+08]                SS:03EC=0001
-
```

Stack için 512 Wordlük yer ayrılmıştır.

Stack'e atma işlemi sondan başlar.

SP=03E4 degerini gösteriyor ($400 - 03E4 = 1C$) yani 14 Wordlük stacke deger atılmıştır. Sondan başlayarak sırasıyla:

PUSH DS; Programın geri dönüş adreslerini. Yığın olarak tanımlanmış kesimde saklar 07 5A

PUSH AX; Programın geri dönüş adreslerini. Yığın olarak tanımlanmış kesimde saklar 00 00

PUSH AX; 1. dizinin 2. elemanı 00 18

PUSH AX; 2. Dizinin 2. elamanı 00 02

CALL RECUR; Yordam çağrıldı kalınan yerden 1 sonraki adımın adresi stacke atılır 00 1B

PUSH BP; BP stacke atılır çünkü Stackten deger çekme için BP kullanılır 00 00

PUSH BX; 00 01

PUSH DX; 00 00

PUSH DX; 00 30

PUSH BX; 00 01

CALL RECUR 00 61

PUSH BP; BP stacke atılır çünkü Stackten deger çekme için BP kullanılır 03 F0

PUSH BX; 00 01

PUSH DX; 00 30

c) Kod yürütülmesi tamamlandıktan sonra veri kesiminin son halini yazınız.

Veri kesiminin ilk hali :

Stack'i 512 Word olarak tanımladığım için ds adres degerleri örnekte verilene göre farklı oldu.

```
-d ds:0
07AA:0000  0D 18 1F FB 28 0B 01 34-14 FF 01 02 03 1B 0A 0A
07AA:0010  FF 19 42 FF 00 00 00 00-00 00 00 00 00 00 00 00
07AA:0020  00 00 00 00 00 00 00 00-0A 00 00 00 00 00 00 00
```

```
-d ds:0
07AA:0000  0D181FFB280B0134-14FF0102031B0A0A
07AA:0010  FF1942FF00000000-0000000000000000
07AA:0020  0000000000000000-0A00000000000000
```

Veri kesiminin son hali :

```
Program terminated normally
-D DS:0
07AA:0000  0D 18 1F FB 28 0B 01 34-14 FF 01 02 03 1B 0A 0A
07AA:0010  FF 19 42 FF 0D 00 30 00-5D 00 79 1A 90 01 6E 00
07AA:0020  FF 00 14 05 28 05 01 FE-0A 00 00 00 00 00 00 00
```

```
-d ds:0
07AA:0000  0D181FFB280B0134-14FF0102031B0A0A
07AA:0010  FF1942FF0D003000-5D00791A90016E00
07AA:0020  FF001405280501FE-0A00000000000000
```