

datasg SEGMENT PARA 'veri'

a DB 12

b DB 17

c DB 12

tip DB ?

datasg ENDS

stacksg SEGMENT PARA STACK 'yigin'

Dw 12 DuP(?)

stacksg ENDS

codesg SEGMENT PARA 'kod'

ASSUME CS:codesg, DS:datasg, SS:stacksg

ANA PROC FAR

*PUSH DS j geri dönüş adreslerinin
ayarlanması*

XOR AX,AX

PUSH AX

*Mov Ax, datasg ! Data segment degerinin
ayarlanması*

Mov DS, AX

Mov AL, 0

Mov BL, b

Mov CL, c

Cmp AL, BL

JE J1

Cmp AL, CL

JE J2

Cmp BL, CL

JE J2

Mov tip, 3

JMP Jon

J1 : Cmp AL, CL

JNE J2

Mov tip, 1

JMP Jon

J2 : Mov tip, 2

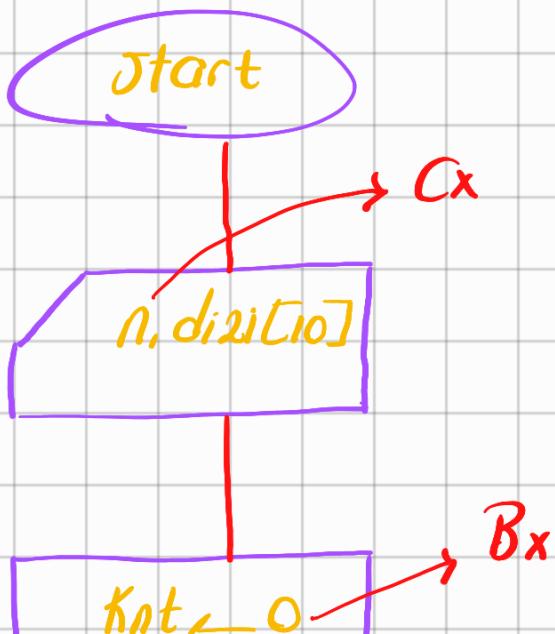
JON RETF

ANA ENDP

Codesg ENDS

END ANA

ÖR → Byte türündeli bir dizinin küçükten büyüğe sıralı olup olmadığını bulunuz



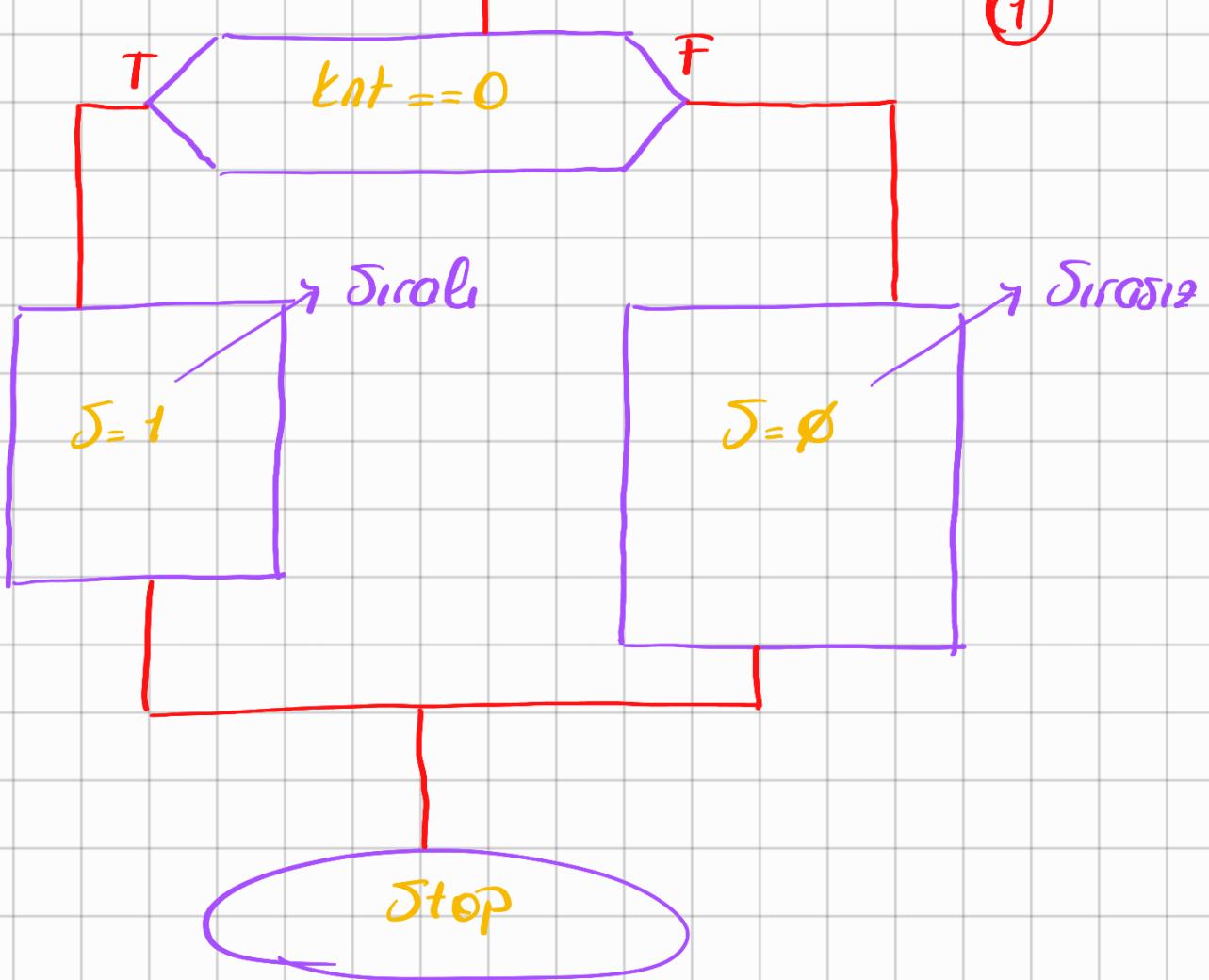
$i \leftarrow 0$ $\rightarrow S1$

$knt = \emptyset \And i < n - 1$

$dis[i][i] > dis[i][i+1]$

$knt \leftarrow 1$

$i = i + 1$



myds SEGMENT PARA 'veri'

dizi DB 1,2,3,4,5

n DW 5

J DB 0

myds ENDS

myss SEGMENT PARA STACK 'yigin'

DW 20 DUP(?)

myss ENDS

mycs SEGMENT PARA 'kod'

ASSUME CS:mycs, SS:myss, DS:myds

main PROC FAR

PUSH DS

XOR AX, AX

PUSH AX

MOV AX, myds

MOV DS, AX

XOR SI, SI ; İndisi Ø'ladık

MOV BX, 0 ; Knt degerine Ø otadık

Mov CX, n ; dönpü sayımasını CX'e otadık

DEC CX ; Dönpü sayımasını 1 azalttık

don : CMP BX, 0 ; knt degistkeni 0'a esit mi
diye kontrol ediyoruz
JNE don_if

CMP SI, CX ; indisimiz ile döngü sayisini
karsilastiririz

JGE don_if

Mov AL, dizi[SI]

CMP AL, dizi[SI+1]

JLE ortir

Mov BX, I

ortir : INC SI

JMP don

don_if : CMP BX, 0

JNE don

Mov SI, I

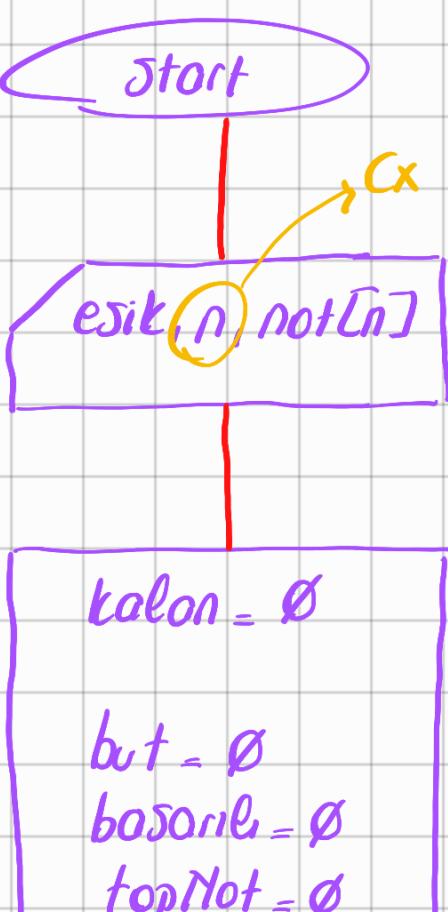
don : RETF

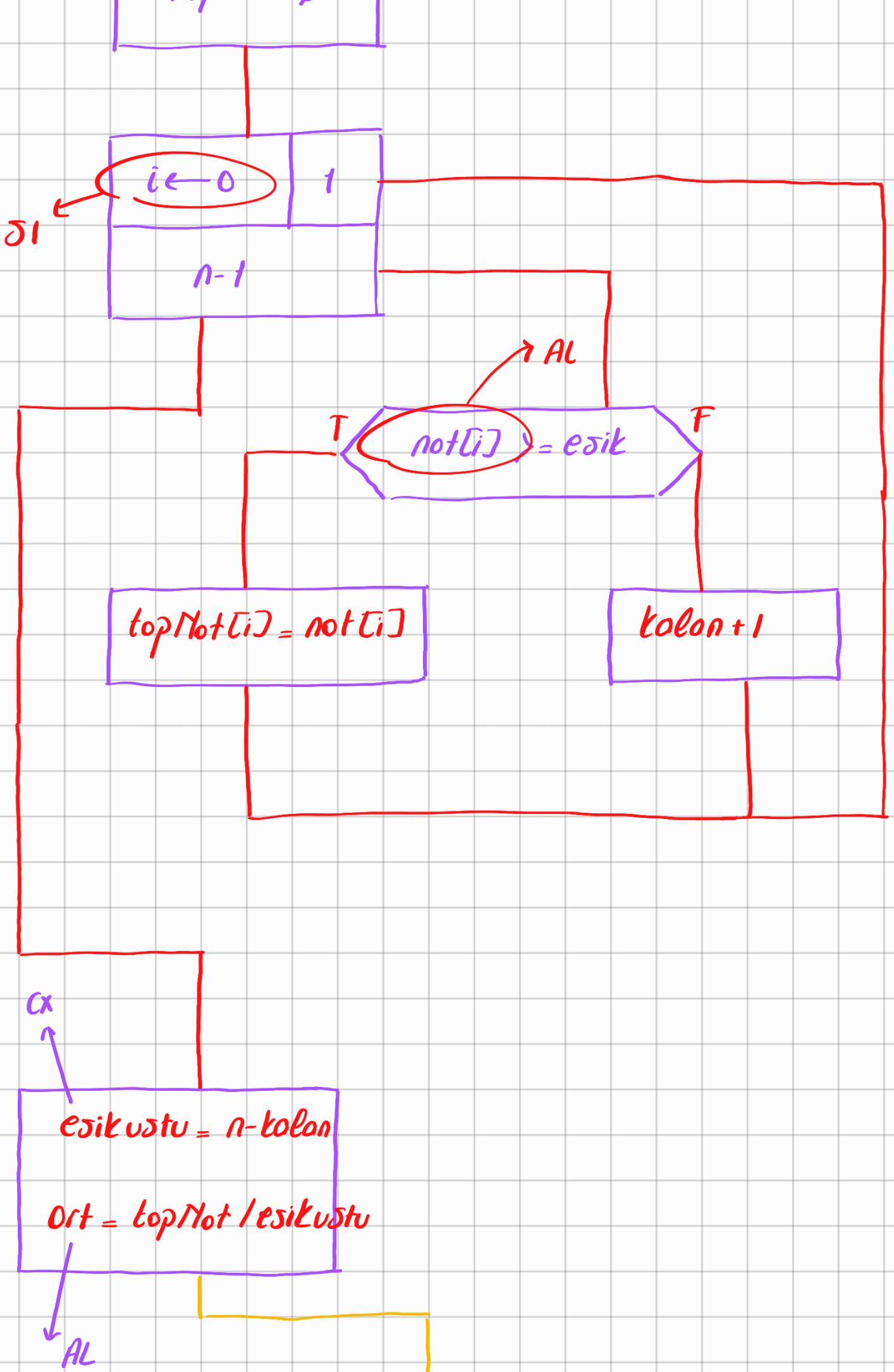
MAIN ENDP

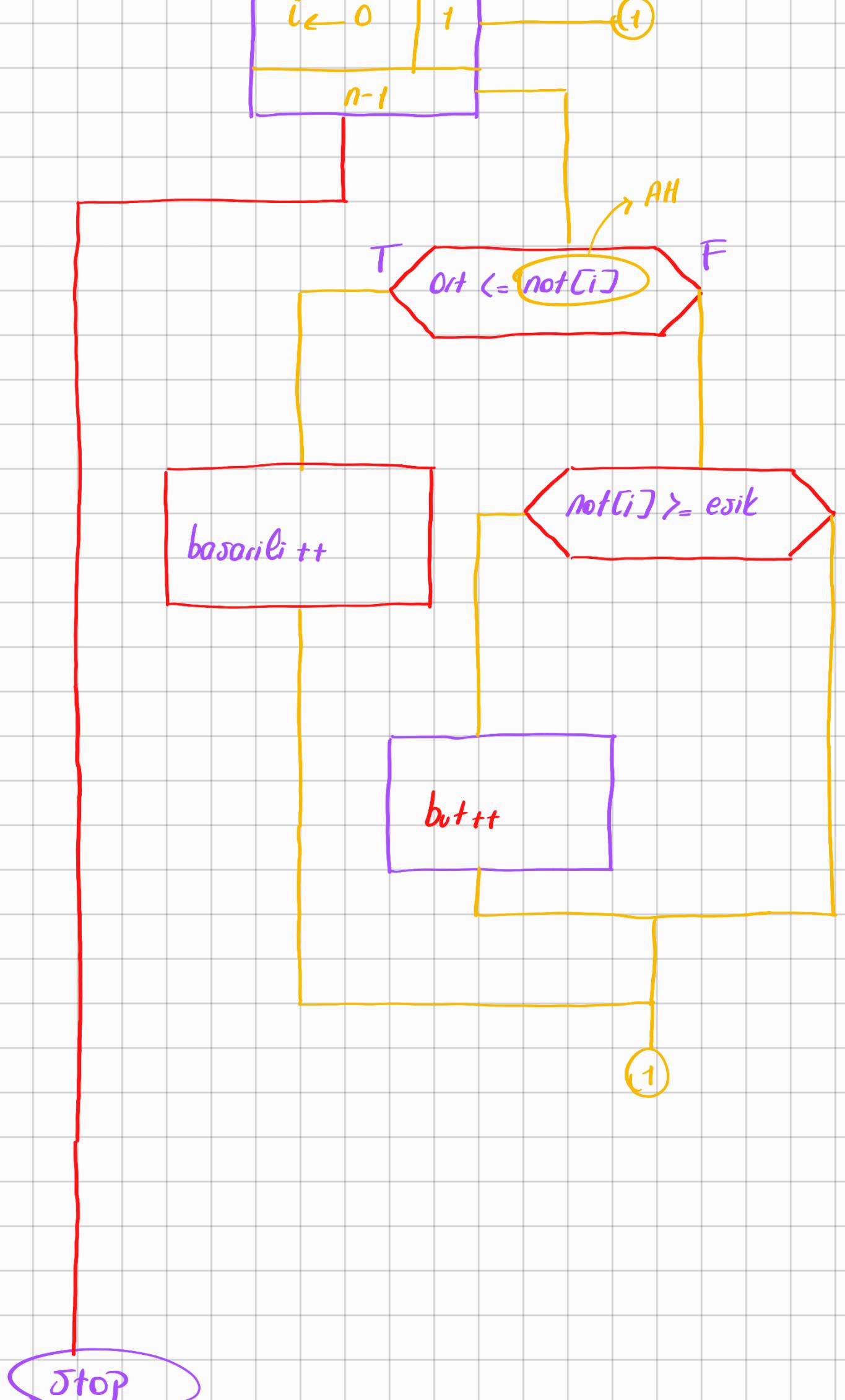


ÖR: Alt serİYE programlAMA dersinin yıl
sonu notları hesaplanmaktır. eşit degerin

Oltinda notu olan öğrencilerin başarısız, esik değerin üstünde notu olan öğrencilerin not ortalaması hesaplanır ve ortolamanın oltında kalan öğrenciler bütünlüğe girmeye hak kazanır. Ortolamanın üstündeki öğrenciler ise dersten başarılı olur. 280 kişisinin aldığı dersten başarılı, başarısız ve bütünlüğe kalan öğrenci sayılarını bulunuz







mystack SEGMENT PARA STACK 'y'

Dw 20 Dup(?)

mystack ENDS

myds SEGMENT PARA 'r'

n Dw 280

kolon Dw 0

but Dw 0

basarili Dw 0

topNot Dw 0

exit DB 40

notlar DB 280 DUP(?)

myds ENDS

MyCS SEGMENT 'E'

ASSUME CS:myCS, SS:myStack, DS:myds

CAN PROC FAR

PUSH DS

XOR AX, AX

Push AX

Mov AX, myds

Mov DS, AX

XOR SI, SI ; indisi Ø'liyoruz

Mor Cx,n j döngü Jöyisini Cx'e otuyoruz

L1: Mor AL, notlar [81]

CMP AL, esik j notu esik değerle karşılaştırıyoruz

JAE esikustu j esitten yüksek ise esikustu
labelina oylanır

INC kolon j esikten düşük ise kolon'ı
bir arttırırız

jmp d1

esikustu: CBw j AL → AX

ADD topNot, AX

d1: INC \$1 j indisimizi bir arttırırız

Loop L1

Mor Cx,n

Sub Cx, kolon

XOR Dx, Dx

Mor Ax, topNot

DIV Cx j Dx : Ax / Cx → Bölüm ! Ax
kolon ! Dx

XOR SI, SI

Mov CX, N

L2: mov AH, notlar[SI]

CMP AL, AH

JBE gecti

CMP AH, esik

JB koldi

INC but

JMP koldi

pecti! INC basorib

koldi! INC SI

Loop L2

RETF

CAN ENDP

myCS ENDS

END CAN

