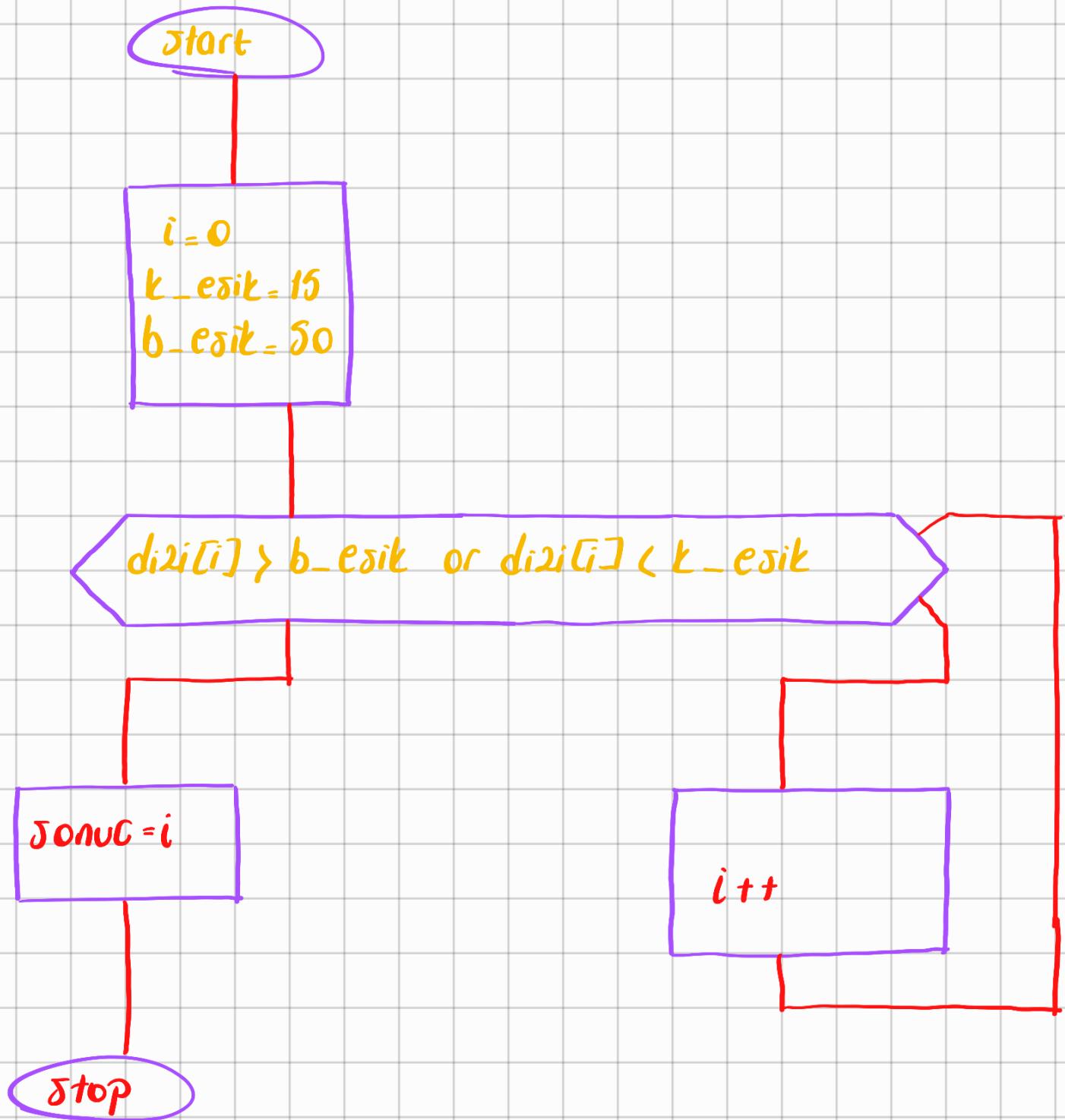


Örnek 2



15 yaşından küçük ve 50 yaşından büyük 10 tane kişinin olduğu sıraya yaş koşulunu bozan 11. kişi karışmıştır. Sıradaki tüm kişinin yaşları bilindipne göre, sıraya sonrasında karışan kişininbastır kaçına sıradan olduğunu bulor. Cm tipi Asm programını yazınız

exe ve Com tipi program karşılaştırması

Özellik

Com program

exe program

Kullanabilen bellek miktarı

64 Kb

Bos bellek ile sınırlı

Boslancıkta CS yazmanın değeri

PSP'nin bulunduğu kesim Adresi

END komutunu takip eden etiket ile belirlenen kodun bulunduğu kesim Adresi

Boslancıkta IP yazmanın değeri

0100H

program bitimini belirleyen END komutunu takip eden etiket ile belirlenen göreli konum adresi

Boslancıkta DS yazmanın değeri

PSP'nin bulunduğu DS kesim adresi

Dönüş için gerekli olan kesim adresi

Tanımlı depistenkere erişmek için yapılması gerekenler

yok

mov Ax, verikesimi

Depistenkelerin tanımlanıldığı yerler

Kesim içinde - ismi mov DS, AX
jmp komutu ile ana yardım orasında reyo

~ veri kesimi

ENDP komutundan
sonra ENDS'den
önce olmot üzere
2 ayrı şekilde

programı bitirmek
için

RET veya
INT21h fonksiyon
4CH

eger dönüş adres-
leni ana yordam
icerisinde yipine
oturmuşsa RETF
olası hallerde
INT 21h fonksiyon
4CH

Dosyanın ikincil
bellet üzerinde-
deti büyütüğü

programın byte
cinsinden karşılığı
kadar

programın boyutu
+ 512 byte exe
header

görsme önceliği exe'den önce

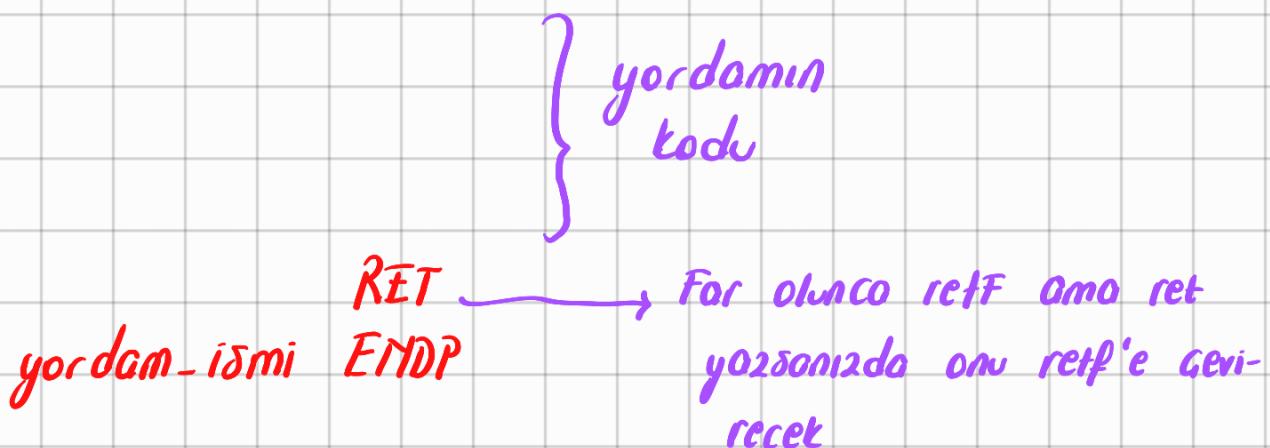
exe'den sonra

Yordam ve macro kullanımları

Birden fazla yordam kullanmak istiyorsak C'yi
düşünün fonksiyon tanımlayıp fonksiyon kullan-
abiliyoruz bir main fonksiyonumuz var bu main

fonksiyondan çağırıldığımız bir gizli fonksiyon
olabiliyor. Bu fonksiyonlara 2 tane 2 tane
parametre aktarılabilir gerekliyor

yordam-ismi PROC {near|far}

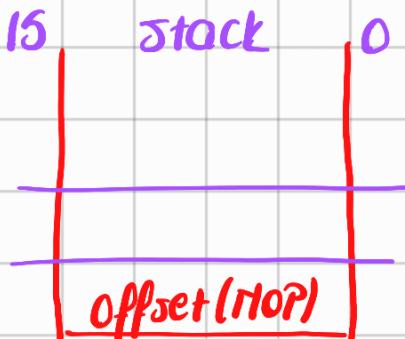


Code seg Segment para 'kod' → Bunu hem exe tipi program
hem com tipi program için
şöylediyebiliriz

ustal Proc Near

ret
ustal endP
Ano proc far

coll ustal



→ coll yordamı çağırırmak için bizim
kullandığımız mnemonic

Mop → hangi yordamı
çağracağımızın onun
adını yazıyoruz
retF
Ano EndP

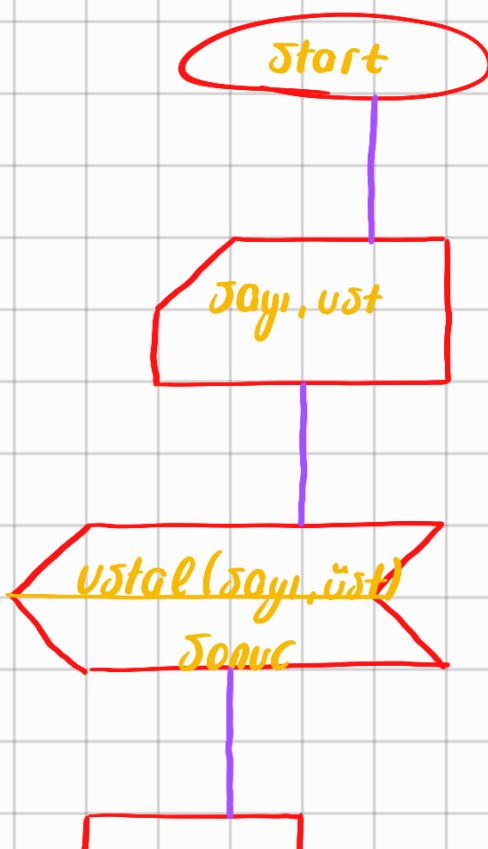
Dolayısıyla coll komutundan

codesg ENDS
END Ano

bir sonraki satırın offset adresini stack'in içine atıyor. Çünkü ustal yordamı çalışıp bittiğinde ret ile beraber bu nöp'te dönüyor olucak, yani ret'e geldiğinde stack'ten o değeri alıyor ve o değeri siliyor. Sonra program noktası itibarı ile sonlandırılıyor.

Örnek 1 - intra-segment yordam

Sayı ve üst değerlerini ana yordamdan ustal yordamına kullanır. Ana yordamı ve üst alma işlemi gerçekleştiriren ustal yordamını exe tipinde yazınız.



Sonuc

Jtop

myss Segment para stack 'yigin'

Dw 20 Dup(?)

myss ends

myds Segment para 'veri'

Sayi Dw 2

Ust Dw 10

Sonuc Dw ?

myds ENDS

mycs Segment para 'kod'

Assume CS:myCS, DS:myDS, SS:mySS

Ana proc far

Push DS

XOR AX, AX

Push AX

Mov AX, myds

Mov DS, AX

}

Standart

oluçak olan

Kod

Mov BX, Sayi

Mov CX, Ust

Coll ustol

Mov Sonuc, AX

RET

Ana EndP

fstl proc near

mov Ax

mul Bx

Loop L1

RET

fstl endP

mycs Ends

End Ana

Örnek 2 inter-segment yordam

BL ve BH üzerinden aldığı 2 deperi toplayın ve Ax
Üzerinden sonucu döndüren horici yordamı ve bu
horici yordamı gizlənən ana yordamı exe tipinde
yazınız

extrn toplama ;Far

myss Segment para stack 'yipin'

Dw 20 DuPL(?)

myss ends

myds Segment para 're'

Sayı1 DB 17

Sayı2 DB 29

Sonuc Dw ?

myds ENDS

mycs Segment para 'to'

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

Ana proc for

Push DS

XOR Ax, Ax

Push Ax

Mov Ax, myds

Mov DS, Ax

Mov BL, sayi1

Mov BH, sayi2

Call toplama

Mov Sonuc, Ax

RETF

Ana EndP

mycs ENDS

END Ana

Public Toplama
Codes Segment para 'kod'

Assume CS:Codes

Toplama Proc For

XOR Ax, Ax

Mov AL BL

Bu harici yordam olucapi
igin extradon kendi
data segmentine, stack
segmentine ihtiyaci olma-
digi igin sadece code
segmentini tanimliyoruz

MOV AL, DL
ADD AL, BH
ADC AH, 0
RETF

toplama EndP
codes Ends
END

Matro tanımı (syntax)

macro_ismi Macro { parametreler }

Local

Assume

----- } matro
----- kodu

EndM

matro oldugunu

belli eden pseudo
komutu yaziyoruz

eger bizim ona
yordom icerisinde

kullandigimiz labellar

varsa o labellarin aynisini
burda kullanmak istiyorsak
ve matro'yu duplicate bir

sekilde kullandigimizda
her seferinde aynı labello

fonsiyon n kez kullanildiginda tek bir olan üzerinde ziplamat
oma makronun her yazildiginda matro kopya- istemiyorsak
lanip yapistiriliyor

Local'le beraber
o labellori tonim-
liyo olucuz

Örnek 3 - macro

n elemanlı bir dizinin elementlerini 2'ye bolen main yordamın

İçinde dizinin en büyük elemanını bulan macro'yu com tipi program için yazınız

Cseg Segment para 'ortak'

ORG 100h

Assume CS:Cseg, SS:Cseg, DS:Cseg

enkck macro dizi,n

XOR SI,SI

Mov AL,dizi[SI]

INC SI

Mov CX,N

Dec CX

L1: cmp al,dizi[SI]

JL don

Mov AL,dizi[SI]

don: inc SI

Loop L1

Endm

Ano proc near

XOR SI,SI

Mov CX,N

L1: Sar dizi[SI],1

INC SI

Loop L1

enkck dizi,n

mor kck, AL

RET

Ano endP

dizi DB 10, -2, 4, 6, 10, 3, 5, 17

n Dw 8

kck DB ?

csep Ends

End Ano

