

# **YILDIZ TEKNİK ÜNİVERSİTESİ**

**ELEKTRİK ELEKTRONİK FAKÜLTESİ**

**BİLGİSAYAR MÜHENDİSLİĞİ**

**ALT SEVİYE PROGRAMLAMA**

**ÖDEV-1**

**Ödevi Yapan Öğrenci**

Onur KOÇKAN

17011701

**Ödev Teslim Tarihi**

06.12.2019

## SORU 1

```
void SteganografiBul(int n, int resim_org, int resim_ste, int steganografi_adres) {  
    __asm {  
        XOR EAX,EAX //EAX SIFIRLANIR  
        XOR DX,DX //DX SIFIRLANIR  
        XOR EBX,EBX //EBX SIFIRLANIR  
        MOV ECX, n //DIZI BOYUTUNUN(WIDTHXHEIGHT) DEĞERİ ECX'E TAŞINMASI,AYNI  
        ZAMANDA L1 LOOPU BU SAYI KADAR DÖNER  
        XOR ESI,ESI //ESI SIFIRLANIR  
  
        L1:  
            PUSH ECX //CX YAZMACI KULLANILMAK İSTENDİĞİNDEN DOLAYI DONGUYU  
            BOZMAMASI İCİN STACK'E ATILIR  
            MOV EBX, resim_org // EBX REGISTERINA GELEN PAREMETREDEKİ DİZİNİN  
            BASLANGIC ADRESİ ATILIR  
            MOV CX, WORD PTR[EBX + EAX] //CX REGISTERINA resim_org ADRESİNDE  
            EAX İNDİSİNDEKİ DEĞER ATILIR  
            MOV EBX,resim_ste//EBX REGISTERINA resim_ste DİZİNİN ADRESİ  
            ATILIR  
            MOV DX, WORD PTR [EBX+EAX] // BU ADRESDEKİ EAX REGISTERINDAKİ  
            DEĞERİN İNDİSİ ATILIR  
  
            CMP DX, CX //ŞUAN ELDE OLAN İKİNCİ DİZİDEKİ PIXELİN DEĞERİNİN  
            BİRİNCİ DİZİDEKİ PIXELİN DEĞERİYLE KARSILAŞTIRILMASI İŞLEMİ  
            JZ DEVAM //EĞER FARK YOK İSE DEVAM LABELINA GİDİLSİN(ZF=1 OLURSA)  
            JS NEGATIF //CXIN DXDEN BÜYÜK OLMA DURUMU (COMPARE ÇIKARMA  
            İŞLEMİNDE NEGATİF SONUC ELDE EDİLİRSE SIGN FLAG=1 OLUR BU DEMEK OLUYORKİ CX,DXDEN  
            BÜYÜK  
  
            MOV EBX,steganografi_adres //(BU DURUMUN İŞLENMESİ=DX CXDEN BÜYÜK  
            VE DEĞERLERİ FARKLI)EBX REGISTERINA CHAR TIPİNDEKİ DİZİNİN BASLANGIC ADRESİ ATILIR  
            SUB DX, CX //DX-CX FARKI ALINIR DX'E ATILIR  
            MOV WORD PTR[EBX+ESI], DX //BULUNAN FARK ŞİFRE DİR VE BU ŞİFRE  
            GELEN PAREMETREDEKİ CHAR DİZİSİNİN ESI.Cİ İNDİSİNE ATILIR  
            ADD ESI,1 //CHAR DİZİSİ OLMASINDAN SEBEB BİR SONRAKİ İTERASYON  
            İÇİN ESI 1 ARTTIRILIR  
            JMP DEVAM //CXIN DXDEN BÜYÜK OLMA DURUMU ATLANILSIN VE İTERASYONA  
            DEVAM EDİLSİN (DEVAM LABELINA GİDER)  
  
            //CXIN DXDEN BÜYÜK OLMA DURUMU  
            NEGATIF:MOV EBX, steganografi_adres//EBX YAZMACINA CHAR TIPİNDEKİ SİFREYİ TUTAN  
            DİZİNİN BASLANGIC ADRESİ ATILIR  
            SUB DX, CX //0-255DEN SONRA TAŞMA OLUCAGI DXE CX EKLENİR  
            ADD DX,256 //VE BU DEĞER 255DEN TAŞTIRILIR  
            MOV WORD PTR[EBX+ESI], DX //BİR ÜST SATIRDAKİ DXE İŞLENEN(DX+256)  
            DEĞER ARADIĞIMIZ ŞİFRENİN KARAKTERİDİR.BU DEĞERDE CHAR TIPİNDEKİ DİZİNİN ESI.Cİ  
            İNDİSİNE ATILIR  
            ADD ESI,1 //İTERASYON İCİN ESI 1 ARTTIRILIR(CHAR TIPİNDE  
            OLDUGUNDAN DOLAYI)  
            DEVAM:  
                //EŞİT OLMA DURUMU  
                ADD EAX,2 //EAXİE İTERASYON İÇİN 2 EKLER  
                POP ECX //YIĞINA ATILAN ECX TEKRAR ÇEKİLİR  
                LOOP L1 //ECX=0 OLANA KADAR DONGU DEVAM EDER  
  
                //OGRENCİ NUMARASI KARAKTER KARAKTER SIRAYLA BELİRTİLDİĞİ ŞEKİLDE  
                DİZİYE EKLENİR  
                MOV EBX, steganografi_adres //ŞİFREYİ TUTAN CHAR TIPİNDEKİ DİZİNİN  
                BAŞLANGIÇ ADRESİNİ EBX REGISTERINA ATAR
```

```

MOV WORD PTR[EBX+ESI], ' ' //TIRNAK İÇİNDEKİ KARAKTERLERİ
SIRASIYLA(ESI+1,ESI+2,.....,ESI+10)ŞİFREYİ TUTAN DİZİYE ATAR(98-108.SATIRLAR)
MOV WORD PTR[EBX + ESI+1], '-'
MOV WORD PTR[EBX + ESI+2], ' '
MOV WORD PTR[EBX + ESI+3], '1'
MOV WORD PTR[EBX + ESI+4], '7'
MOV WORD PTR[EBX + ESI+5], '0'
MOV WORD PTR[EBX + ESI+6], '1'
MOV WORD PTR[EBX + ESI+7], '1'
MOV WORD PTR[EBX + ESI+8], '7'
MOV WORD PTR[EBX + ESI+9], '0'
MOV WORD PTR[EBX + ESI+10], '1'
}
}

```

## SORU 2

Yapılış Raporu: Kullanıcıdan girilen aday kenarlar ile oluşturulabiliniyorsa oluşan en küçük çevreye sahip olan üçgeni bulmak.

- 1-KULLANICI KAÇ ADET ADAY KENARI GİRİCEĞİNİ BELİRLER
- 2-BU VERİLEN ADET ŞARTLARA UYGUNMU KONTROLÜ YAPILIR UYGUN İSE;
- 3-KULLANICIDAN KENARLAR,KENDİSİNİN BELİRTİLDİĞİ KADAR DEFA ALINIR
- 4-İLK ÖNCE DİZİ KÜÇÜKTEN BÜYÜĞE SIRALANIR BUNUN AMACI BULUNAN İLK ÜÇGEN DİĞER BULUNABİLİNECEK ÜÇGENLERDEN DAHA AZ ÇEVREYE SAHİP OLSUN.
- 5-DAHA SONRA KÜÇÜKTEN BÜYÜĞE SIRALANAN DİZİ İÇİN ÜÇGEN ŞARTI
  - A=İLK KENAR
  - B=İKİNCİ KENAR
  - C=ÜÇÜNCÜ KENAR
  - I)A+B>?C
  - II)C-B<?A
- ŞARTI SAĞLANIYOR İSE ÜÇGEN OLUŞUR
- 6-OLUŞAN ÜÇGENİN A,B VE C DEKİ DEĞERLERİ KULLANICIYA “ÜÇGEN OLUŞTU” BİLGİSİYLE GÖSTERİLİR.

## KODLAR VE AÇIKLAMALARI:

dataSeg SEGMENT PARA 'veri' ;dataSeg ISMINDEKİ DATA SEGMENTİN BULUNDUGU BLOK VE TANIMLAMASI  
MSG1 DB 'KENAR DİZİSİNİN ELEMAN SAYISINI GIRIN:',0 ;(BYTE BOYUTUNDA) KULLANICIDAN KENAR SAYISI ALINMADAN ÖNCE YAZDIRILAN MESAJIN TANIMLAMASI  
MSG2 DB CR,LF,'KENAR GIRIN: ',0 ;(BYTE BOYUTUNDA) KULLANICIDAN ADAY KENAR ALINMADAN ÖNCE YAZDIRILAN MESAJIN TANIMLAMASI  
MSG3 DB CR,LF,'GİRİLEN ADAY KENARLARIN DİZİ SAYISI EN AZ 3-EN ÇOK 100 ARALIGINDA OLMALIDIR!',0 ;(BYTE BOYUTUNDA) KULLANICININ GİRDİĞİ KENAR DİZİSİ SAYISI  
;İSTENİLEN ARALIGIN DISINDA İSE  
;YAZDIRILAN MESAJIN TANIMLAMASI  
MSG4 DB CR,LF,'->',0 ;(BYTE BOYUTUNDA)KENARLARI YAZDIRIRKEN '-' SEMBOLUYLA DAHA BELİRGİN HALE GETİRİLMESİ İÇİN KULLANILAN DEĞİSKENİN TANIMLAMASI  
MSG5 DB CR,LF,'GİRİLEN ADAY KENARLARIN DEĞERİ EN AZ 3-EN ÇOK 1000 OLMALIDIR!',0 ;(BYTE BOYUTUNDA)KULLANICININ GİRDİĞİ ADAY KENAR SAYISI ARALIGIN DISINDA İSE  
;YAZDIRILACAK METİNİN TANIMLAMASI  
MSG6 DB CR,LF,'ÜÇKEN KURULDU KURULAN ÜÇGENİN KENARLARI:',0 ;(BYTE BOYUTUNDA)ADAY KENARLARDAN ÜÇGEN KURULMA SARTI OLUSTUGU ZAMAN KENARLARDAN ÖNCE YAZDIRILACAK METİNİN TANIMLAMASI  
MSG7 DB CR,LF,'ÜÇGEN KURULABİLECEK ADAY KENAR YOK!',0 ;(BYTE BOYUTUNDA)ADAY KENARLARDAN ÜÇGEN KURULMA SARTI OLUSMUYORSA YAZDIRILACAK METİNİN TANIMLAMASI  
K1 DW ? ;(WORD BOYUTUNDA)YAZDIRMA İŞLEMİNDEN ÖNCE VE KONTROL İŞLEMLERİNDE REGISTERLARDAKİ İFADEYİ YEDEKLEMELİK AMACI İLE TUTULAN 1.KENARI TEMSİLEN OLUSTURULAN DEĞİSKEN  
K2 DW ? ;(WORD BOYUTUNDA)YAZDIRMA İŞLEMİNDEN ÖNCE VE KONTROL İŞLEMLERİNDE REGISTERLARDAKİ İFADEYİ YEDEKLEMELİK AMACI İLE TUTULAN 2.KENARI TEMSİLEN OLUSTURULAN DEĞİSKEN  
K3 DW ? ;(WORD BOYUTUNDA)YAZDIRMA İŞLEMİNDEN ÖNCE VE KONTROL İŞLEMLERİNDE REGISTERLARDAKİ İFADEYİ YEDEKLEMELİK AMACI İLE TUTULAN 3.KENARI TEMSİLEN OLUSTURULAN DEĞİSKEN  
N DW 100 ;(WORD BOYUTUNDA)DİZİNİN ELEMAN SAYISINI TUTAN DEĞİSKEN.GECİCİ OLARAK 100 VERİLMİŞTİR  
KOD İÇERİSİNDE KULLANICIDAN KONTROLLERİ VE DİZİNİN ELEMAN SAYISININ DEĞERİNİ ALMA  
;VE TUTMA İŞLEMLERİNDE YENİ(ASİL) DEĞERİNE SAHİP OLACAKTIR.  
DİZİ DW 100 DUP(?) ;(WORD BOYUTUNDA)ADAY KENAR ÜÇGENLERİNİN TUTULACAKI DİZİ  
;ALT SATIRA GEÇMEK İÇİN KULLANILAN DEĞİSKENLER(İn İSLEVİ GİBİ)  
CR EQU 13  
LF EQU 10  
HATA DB CR,LF,'DİKKAT!SAYIDAN HARİC BASKA BİR GİRİŞ YAPMAYIN',0 ;SAYIDAN HARİC BASKA BİR INPUT DEĞERİ OKUNURSAKİ ÇIKACAK MESAJIN  
dataSeg ENDS ;DATA SEGMENTİN BİTİSİ

stackSeg SEGMENT PARA STACK 'yigin' ;stackSeg STACK SEGMENTİN BULUNDUGU BÖLÜM VE TANIMLAMASI  
DW 32 DUP(?) ;32 WORDLUK STACKİN TANIMLAMASI  
stackSeg ENDS

codeSeg SEGMENT PARA 'kod' ;codeSeg İSMİNDE CODE SEGMENTİN BAŞLADIĞI SATIR  
ASSUME CS:codeSeg,DS:dataSeg,SS:stackSeg ;SEGMENTLERİN İSİMLERİNİN TANITILMASI

BASLA PROC FAR ;EXE TİPİ PROGRAMLARDA GEREKLİ OLAN BİR ADET FAR TİPİNDEKİ ANA PROSEDÜRÜN(BASLA İSMİNDE) BAŞLANGICI  
;DONUŞ İÇİN GEREKLİ OLAN DEĞERLERİ YIGINDA SAKLAMA İŞLEMİ  
PUSH DS  
XOR AX,AX  
PUSH AX  
;dataSeg İSMİYLE TANIMLI KESİM ALANINA ERİŞEBİLMEK İÇİN GEREKLİ TANIMLAMA  
MOV AX,dataSeg  
MOV DS,AX

;ODEV SORU:2NİN ÇÖZÜMÜNÜN BAŞLADIĞI ASİL YER.

;İLK OLARAK KULLANICIDAN BİR N DEĞERİ ALINIR 3 İLA 100 ARASINDAMI DİYE KİYASLANIR(N=DİZİNİN BOYUT SAYISI)

XOR SI,SI  
MOV AX,OFFSET MSG1  
CALL PUT\_STR;MSG1İ GÖSTEREN PROSEDÜRÜN ÇAĞIRILMASI  
CALL GETN;SAYIYI OKUYAN PROSEDÜRÜN ÇAĞIRILMASI  
MOV N,AX ;PROSEDÜR DONUŞU ALINAN SAYI AXDE TUTULUR.BU DEĞERİ N(DİZİNİN ELEMAN SAYISI)E ATILMASI.

CMP N,3 ;N 3DEN BÜYÜKMU  
JAE CONT1 ;BÜYÜK İSE 1000DEN KÜÇÜKMU VE ESİTİMİ DİYE KARŞILAŞTIR  
MOV AX,OFFSET MSG3 ;DEĞİL İSE MSG3U AX E AT  
CALL PUT\_STR ;AXDEKİ DEĞERİN EKRAHA YAZILMASI  
JMP SON ;SARTLAR UYUŞMADI.SON LABELINA ZIPLA

CONT1: CMP N,100 ;N 1000DEN KÜÇÜKMU VE ESİTİMİ  
JBE CONT2 ;KÜÇÜK VE ESİT İSE DİZİYE DEĞER ALMA İŞLEMİNE BAŞLAYAN LABELA(CONT2) ZIPLA  
MOV AX,OFFSET MSG3  
CALL PUT\_STR  
JMP SON

```

;KENARLARIN DIZISINE DEGERLERI ALMA
CONT2:
LEA SI,DIZI
MOV CX,N ;N(DIZININ ELEMEN SAYISI)
;DIZININ ELEMEN SAYISI KADAR DONGUYE GIR VE DEGER AL
L1:MOV AX,OFFSET MSG2
CALL PUT_STR
CALL GETN ;KLAVYEDEN OKUNAN DEGER AXDE TUTULUR

CMP AX,0 ;ALINAN ILK ADAY KENARIN DEGERI 0DAN BUYUK OLMALIDIR
JA CONT3 ;BUYUK ISE IKINCI SARTI KONTROL EDEN LABEL'A(CONT3) ZIPLA
MOV AX,OFFSET MSG5 ;SARTLAR UYUSMADI.SON LABELINA ZIPLA
CALL PUT_STR
JMP SON

CONT3:CMP AX,1000 ;IKINCI KONTROL GIRILEN SAYI 1000E ESIT VE KUCUKMU
JBE CONT4 ;KUCUK VE ESIT ISE DEGERI DIZININ SI.CI INDISINE AL (CONT4 LABELINA ZIPLA)
;GIRILEN KENAR 2 SARTADA UYMADIGI ICIN CX INDISI BIR AZALTILIP TEKRAR DONGUYE GIRER(SON ADIMI
TEKRARLAR
MOV AX,OFFSET MSG5
CALL PUT_STR
INC CX
LOOP L1

CONT4:MOV [SI],AX
ADD SI,2 ;SI'NIN DEGERINI 2 ARTTIR(SONRAKI INDISDEKI DEGERE ULASMAK ICIN)
LOOP L1

;KENARLARIN DIZISINI KUCUKTEN BUYUGE SIRALAMA ISLEMI YAPILIYOR CUNKU BULUNAN ILK UCGENDE
PROGRAM TAMAMLANICAK VE EN KUCUK CEVREYE SAHIP OLAN UCGEN BULUNMUS OLUNACAK
;CX'E DIZININ ELEMEN SAYISI ATILIR VE 1 EKSIGI KADAR ITERASYON YAPILIR
MOV CX,N
DEC CX
;BUBBLESORT ALGORITMASININ YAPILDIGI BOLUMDUR.MEVCUT DEGER AXDE TUTULUR SIRADAKI DEGER DXDE
TUTULUR VE KARSILASTIRILIR
ARA:
MOV BX,CX
XOR SI,SI
KARSILASTIR:
MOV AX,DIZI[SI]
MOV DX,DIZI[SI+2]
CMP AX,DX
;MEVCUT INDISDEKI DEGER SONRAKI DEGERDEN KUCUK ESIT ISE YERINI DEGISTIRME VE DEGISTIRME LABELINA
ZIPLA
JBE DEGISTIRME
;BUYUK ISE BIR SONRAKI INDISDEKI DEGERI MEVCUT INDISDEKI DEGERLE DEGISTIR
MOV DIZI[SI],DX
MOV DIZI[SI+2],AX

;SI INDISINI ARTTIR DONGUYE DEVAM ET
DEGISTIRME:
ADD SI,2
DEC BX
JNZ KARSILASTIR

LOOP ARA

;KUCUKTEN BUYUGE SIRALANAN DIZI ICIN EN KUCUK CEVREYE SAHIP OLAN UCGEN OLMA SARTI
;A+B>C VE C-B<A NIN KONTROLUNE BASLANAN KOD BLOGU

XOR AX,AX ;MEVCUT INDISDEKI 1.KENARI TUTACAK
XOR BX,BX ;MEVCUT INDISDEKI 3.KENARI TUTACAK
XOR DX,DX ;MEVCUT INDISDEKI 2.KENARI TUTACAK

XOR DI,DI ;ITERASYON ICIN TUTULACAK INDISIN SIFIRLANMASI

;DONGUNUN DIZININ BOYUTUNDAN 2 EKSIGI KADAR ILERLEMESI ICIN GEREKLI ATAMALAR

```

```

MOV CX,N
SUB CX,2
;3 KENARINDA SIRALI ITERASYONU ICIN 3 ADET IC ICE LOOP BULUNUR
;ILK LOOP MEVCUT INDISDEKI 1.KENARI TUTAR,2.LOOP 2.KENARI VE 3.LOOP 3.KENARI TUTAR
L4:PUSH CX ;L4 LOOPUNUN DEGERI KAYBEDILMEMESI ICIN STACKE PUSHLANIR
MOV AX,DIZI[DI] ;AX REGISTERINA DIZININ SININCI INDISINDEKI(KENAR 1) DEGERIN ATILMASI
MOV K1,AX ;AX REGISTERINDAKI 1.KENAR DEGERI K1 DEGISKENINE YEDEKLEME AMACI ATILIR
;DIZI ITERASYONU ICIN DI IKI ARTTIRILIR
INC DI
INC DI
PUSH DI ;AX ICIN YAPILAN ITERASYONDAKI INDIS DEGERINI TUTAN DI AYRI OLARAK STACKE ATILIR(HER LOOP VE
KENAR TUTULMASI ICIN DI REGISTERI KULLANILDI CUNKU)
L5: PUSH CX
MOV DX,DIZI[DI] ;2.KENARI DX YAZMACINA ATILMASI
MOV K2,DX ;2.KENARIN YEDEKLENMESI
INC DI
INC DI
PUSH DI
L6:PUSH CX
MOV BX,DIZI[DI] ;ARTTIRILAN DI DEGERINDEKI MEVCUT DEGERI BXE AT(3.KENAR)
MOV K3,BX ;3.KENARIN YEDEKLENMESI
INC DI
INC DI

;A=1.KENAR , B=2.KENAR , C=3.KENAR
; A+B>?C KONTROLUNUN YAPILDIGI KISMIN BASLANGICI
; KAYBEDILMEMESI ICIN (REGISTERLARDA YAPILAN ISLEMLERDEN DOLAYI DEGISEN DEGERLERI)YEDEKLENEN
KENARLARIN REGISTERLARA TEKRAR ATILMASI
MOV AX,K1
MOV DX,K2
MOV BX,K3

;DIZIDEN TASMA VE UCGENIN OLUSMAMA DURUMU KONTROLU(162-167)
;AYNI ZAMANDA DIZIDEKI SON INDISLERINDE KONTROL DURUMUDUR.PROGRAM DAHA ILERSINE GIDEMEZ VE
STACKTEKI BOS DEGERLER ILE ISLEM YAPAMAZ
CMP AX,0
JZ TASTI ;1.KENARIN DEGERI SIFIRMI.SIFIR ISE TASMISTIR VE BU DEGER ILE UCGEN KURULMAZ TASTI
LABELINA ZIPLA
CMP BX,0
JZ TASTI ;2.KENARIN DEGERI SIFIRMI.SIFIR ISE TASMISTIR VE BU DEGER ILE UCGEN KURULMAZ TASTI
LABELINA ZIPLA
CMP DX,0
JZ TASTI ;3.KENARIN DEGERI SIFIRMI.SIFIR ISE TASMISTIR VE BU DEGER ILE UCGEN KURULMAZ TASTI
LABELINA ZIPLA

;A+B>?C KONTROLU
ADD AX,DX
CMP AX,BX
JA DIGERINEGEC
;ILK SARTI SAGLIYAMADI DEVAM LABELINDEN DIGER DURUMA GEC
JMP DEVAM

; C-B<?A KONTROLU
DIGERINEGEC:
MOV AX,K1
MOV DX,K2
MOV BX,K3

SUB BX,DX
CMP BX,AX
JB UCGEN
;VE IKINCI SARTIDA SAGLIYAMADIGI ICIN DEVAM LABELINDEN IP KODU OKUMAYA DEVAM EDER DIGER
DURUMUN KONTROLU BASLAR

DEVAM:
POP CX ;EN ICTEKI(L6) DONGUNUN STACKE ATILAN DEGERI STACKTEN GERI CEKILIR
LOOP L6
POP DI ;2.KENARIN MEVCUT DIZI INDISINI TUTAN DI DEGERI STACKTEN GERI CEKILIR
POP CX ;IKINCI DONGUNUN(L5) STACKE ATILAN DONGU SAYISI DEGERI GERI CEKILIR.
LOOP L5
POP DI ;1.KENARIN MEVCUT DIZI INDISINI TUTAN DI DEGERI STACKTEN GERI CEKILIR
POP CX ;ILK KENARI(EN KUCUK) TUTAN LOOP(L4)ICIN STACKE ATILAN DONGU SAYISININ DEGERI TEKRAR GERI
CEKILIR
LOOP L4

```

```
;TASTI LABELI UCGEN OLUSMAMA SARTI DURUMUNDA ISLER
TASTI:
MOV AX,OFFSET MSG7
CALL PUT_STR
JMP SON ;UCGEN OLMAMA SARTINDAN SONRA PROGRAMIN SONLANMASI ICIN SON DIYE ADLANDIRILAN LABELA
ZIPLAR
```

```
;UCGEN OLUSMA DURUMUNDA EKRANA YAZDIRILMA ISLEMININ YAPILMASI
UCGEN:
MOV AX,OFFSET MSG6 ;UCGENIN OLUSTUGU BILGISININ KULLANICIYA SOYLENDIGI DEGISKENIN AX
REGISTERINA ATILMASI
CALL PUT_STR ;PUT_STR PROSEDURU ILE EKRANA YAZDIRILMASI

MOV AX,OFFSET MSG4 ;BELIRGINLIK ICIN MSG4E TANIMLADIGIMIZ '-' ISARETINE SAHIP DEGISKENIN AXE ATILIP
ILGILI PROSEDURU CAGIRIP EKRANA YAZDIRILMASI ISLEMI
CALL PUT_STR

MOV AX,K1 ;ILK KENARI TUTAN VE YEDEKLEMESINI YAPTIGIMIZ K1 DEGISKENINI AX REGISTERINA ATILMASI
CALL PUTN ;PUTN PROSEDURU SAYISAL DEGERLERI YAZDIRIR VE AX YAZMACINDAKI DEGERI YAZDIRIR

;AYNI ISLEMLERIN 2.KENAR ICIN GERCEKLENMESI
MOV AX,OFFSET MSG4
CALL PUT_STR

MOV AX,K2
CALL PUTN

;UCUNCU KENARIN EKRANA YAZDIRILMASI
MOV AX,OFFSET MSG4
CALL PUT_STR

MOV AX,K3
CALL PUTN

JMP SON ;PROGRAMIN SONLANMASI AMACI ILE OLUSTURULMUS SON LABELINA ZIPLANILIR

BASLA ENDP
```

```
GETC PROC NEAR;KLAVYEDEN BASILAN KARAKTERI AL REGISTERINA ALIR VE EKRANDA GOSTERIR.ISLEM
SONUCUNDA SADECE AL ETKILENIR
MOV AH,1H
INT 21H
RET
GETC ENDP
```

```
PUTC PROC NEAR;AL REGISTERINDAKI DEGERI EKRANDA GOSTERIR.DL VE AH DEGISIYOR.AX VE DX
REGISTERLARININ DEGERLERI KORUMAK ICIN PUSH/POP YAPILIR
PUSH AX
PUSH DX
MOV DL,AL
MOV AH,2
INT 21H
POP DX
POP AX
RET
PUTC ENDP
```

```
GETN PROC NEAR ;KLAVYEDEN GIRILEN SAYIYI OKUR AX REGISTERI UZERINDE DONDURUR.
;DX:SAYININ ISARETLI OLUP/OLMADIGINI BELIRLER.1(+),-1(-)DEMEK
;BL:HANE BILGISINI TUTAR
;CX:OKUNAN SAYININ ISLENMESI SIRASINDAKI ARA DEGERI TUTAR.
;AL:KLAVYEDEN OKUNAN KARAKTERI TUTAR
;AX ZATEN DONUS DEGERI OLARAK DEGISMEK DURUMUNDADIR.ANCAK DIGER REGISTERLARIN DEGERLERI
ESKISINE DONER
PUSH BX
PUSH CX
PUSH DX
GETN_START:
MOV DX,1 ;SAYIYI SIMDILIK POZITIF OLDUGU VARSAYILIR
XOR BX,BX ;OKUMA YAPMADIGI HANE 0 OLUR
XOR CX,CX ;ARA TOPLAM DEGERI DE 0DIR
NEW:
```

```

CALL GETC ;KLAVYEDEKİ İLK DEĞERİ AL'YE OKU
CMP AL,CR
JE FIN_READ ;ENTER TUSUNA BASILMIŞ İSE OKUMA BİTER
CMP AL,'-' ;AL,'-'Mİ GELDİ KONTROLU
JNE CTRL_NUM;GELEN 0-9 ARASINDA BİR SAYIMI KONTROLU
NEGATIVE:
MOV DX,-1 ;- BASILDI İSE SAYI NEGATİF,DX=1 OLUR
JMP NEW ;YENİ HANEYİ AL
CTRL_NUM:
CMP AL,'0' ;SAYININ 0-9 ARASINDA OLDUGUNU KONTROL ET
JB ERROR
CMP AL,'9'
JA ERROR ;DEĞİL İSE HATA MESAJI VERİLSİN
SUB AL,'0' ;RAKAM ALINDI HANEYİ TOPLAMA DAHİL ET
MOV BL,AL ;BL'YE OKUNAN HANEYİ KOY
MOV AX,10 ;HANEYİ EKLERKEN *10 YAPILACAK
PUSH DX ;MUL KOMUTU DX'İ BOZAR İSARET İÇİN SAKLANIR VE STACKE ATILIR
MUL CX ;DX:AX = AX*CX İŞLEMİ
POP DX ;İSARETİ GERİ AL
MOV CX,AX ;CXDEKİ ARA DEĞER *10 YAPILDI
ADD CX,BX ;OKUNAN HANEYİ ARA DEĞERE EKLE
JMP NEW ;KLAVYEDEN YENİ BASILAN DEĞERİ AL
ERROR:
MOV AX,OFFSET HATA
CALL PUT_STR ;HATA MESAJINI GOSTERİR
JMP GETN_START ;O ANA KADAR OKUNANLARI UNUTTUR VE YENİDEN SAYI ALMAYA BASLATIR
FIN_READ:
MOV AX,CX ;SONUC AX ÜZERİNDEN DONECEK
CMP DX,1 ;İSARETE GÖRE SAYIYI AYARLAMA İŞLEMİ YAPILIR
JE FIN_GETN
NEG AX ;AX=-AX İŞLEMİ YAPILIR
FIN_GETN:
POP DX
POP CX
POP DX
RET
GETN ENDP

```

```

PUTN PROC NEAR ; AX DE BULUNAN SAYIYI ONLUK TABANDA HANE HANE YAZDIRIR
;CX:HANELERİ 10'A BÖLEREK BULACAGIZ,CX=10 OLACAK
;DX:32 BÖLMEDE İŞLEME DAHİL OLACAK.SONUCU ETKİLEMEMESİ İÇİN 0 OLMALIDIR.
PUSH CX
PUSH DX
XOR DX,DX ;DX 32 BIT BÖLMEDE SONUCU ETKİLEMESİN DİYE 0 OLMALI
PUSH DX ;HANELERİ ASCII KARAKTER OLARAK STACKE SAKLAYACAGIZ VE KAC HANENİN ALINACAGINI
BİLMEDİĞİMİZ İÇİN
;STACKE 0 DEĞERİNİ KOYUP ONU ALANA KADAR DEVAM ETTİRİLİR
MOV CX,10 ;CX REGISTERİNE 10 DEĞERİNİN ATILMASI
CMP AX,0
JGE CALC_DIGITS
NEG AX ;SAYI NEGATİF İSE AX POZİTİF YAPILIR
PUSH AX ;AX'İ SAKLA
MOV AL,'-' ;İSARETİ EKRANA YAZDIR
CALL PUTC
POP AX ;STACKTEKİ AX REGISTERİNİN DEĞERİNİN GERİ ALINMASI
CALC_DIGITS:
DIV CX ;DX:AX=AX/CX AX=BÖLÜMÜ TUTAR DX=KALANI TUTAR
ADD DX,'0' ;KALAN DEĞERİNİN ASCII KARŞILIGİNİN BULUNMASI
PUSH DX ;BU DEĞER DX ÜZERİNDEN STACKE ATILIR
XOR DX,DX ;DX SIFIRLANIR
CMP AX,0 ;BÖLEN 0 KALDI İSE SAYININ İŞLENMESİ SONA ERER.
JNE CALC_DIGITS ;İŞLEM TEKRARLANIR
DISP_LOOP: ;YAZILACAK TÜM HANELERİMİZ YIGINDA.EN ANLAMLI HANE ÜSTTE EN AZ ANLAMLI HANE EN
ALTTA VE ONUN ALTINDA DA SONA VARDIĞIMIZI
;ANLAMAK İÇİN KONULAN 0 DEĞERİ VAR.
POP AX ;SIRAYLA DEĞERLER YIGINDAN ALINIR
CMP AX,0 ;AX=0 İSE SONUNA GELİNDİ ANLAMINI TASIR.
JE END_DISP_LOOP
CALL PUTC ;ALDEKİ ASCII DEĞERİ YAZDIRIR
JMP DISP_LOOP ;İŞLEME DEVAM EDİLİR
END_DISP_LOOP:
POP DX
POP CX
RET
PUTN ENDP

```



```
PUT_STR PROC NEAR
;AX DE ADRESI VERILEN SONUNDA 0 OLAN DIZGEYI KARAKTER KARAKTER YAZDIRIR
;BX DIZIYE INDIS OLARAK KULLANILIR.ONCEKI DEGER SAKLANMALIDIR.
PUSH BX
MOV BX,AX ;ADRES BX REGISTERINA ATILIR
MOV AL,BYTE PTR [BX] ;AL DE BYTE POINTERIN BX.CI INDISINI GOSTERDIGI ILK KARAKTER VAR
PUT_LOOP:
CMP AL,0
JE PUT_FIN ;0 GELDI ISE DIZI SONA ERER
CALL PUTC ;AL'DEKI KARAKTER EKRANA YAZDIRILIR
INC BX
MOV AL,BYTE PTR [BX]
JMP PUT_LOOP;YAZDIRMAYA DEVAM EDILIR
PUT_FIN:
POP BX
RET
PUT_STR ENDP
```

;BASLA(MAIN) PROSEDURU ICERISINDE BIRCOK DEFA KULLANILAN SON LABELI PROGRAMI SONLANDIRMA(SONUNA GOTURME/ZIPLAMA)AMACI ILE OLUSTURULMUSTUR  
SON:

;BASLA ADINDAKI FAR TIPINDE OLAN PROSEDURUMUZUN SON BULDUGU YER  
codeSeg ENDS ;codeSeg ISMINDEKI CODE SEGMENTIN BITISI  
END BASLA