

Son Teslim tarihi: 02.01.2022

Ödev 1

OĞUZHAN TOPALOĞLU
Ç19052025 – Grup 3

*Bilgisayar Mühendisliği Bölümü,
Elektrik-Elektronik Fakültesi,
Yıldız Teknik Üniversitesi*



Istanbul, 2021

Başlama Öncesi Notlar:

Bu notları ödevi bitirdikten sonra yazıyorum ve size ödevi yaparken bir sürü şey sormuştum. Bunları not almam gerekirse:

1- İlk altmenünün yalnızca bir kez çalıştırılmasına izin veriyorum.



Oğuzhan Topaloğlu Dün

Hocam selamlar, ödevimi bitirdim ancak bir şey sormak istemiştım. 1. menüde N sayısı alıp N kadar terim ekleyerek diziyi oluşturuyoruz ya, ben bu menünün yalnızca bir kez kullanılabilmesine izin verdim çünkü bu menüyü dizi oluşturma seçeneği gibi düşündüm. Sıkıntı olur mu yoksa menüyü değiştirip menüyü n kez terim ekleme menüsü mü yapmalıyım?



Furkan Çakmak Dün

1 kez oluşturulması yeterli.

2- ÇAP öğrencisi olduğum için numaram Ç19052025 ancak Ç harfi non-ascii karakter olduğundan her yere bilerek 19052025 yazıyorum.



Oğuzhan Topaloğlu 19 Ara 2021

Hocam ben ÇAP öğrencisi olduğumdan numaram Ç19052025 diye geçiyor, dosya adları yüzünden sıkıntı çıkmasın diye 19052025 diye kullansam olur mu?



Furkan Çakmak 19 Ara 2021

Olur

Ödev sırasında sırayla yaptığım şeyler:

MAIN PROSEDÜRÜ:

İlk yaptığım şey exe template'ini kullanarak .asm dosyasının temellerini oluşturmaktı. Bunları yazdıktan sonra kitabımızın github sayfasında bulunan örnekten scan ve print fonksiyonlarını kopyala yapıştır ile aldım ve main prosedürümü oluşturmaya başladım.

Main prosedür kısaca menüyü yazdırıp bir girdi değeri alıyor ve bu değeri AX'te saklıyor. Sakladıktan sonra da AX değerine göre bir altmenüyü çalıştırıyor. Her altmenü için bir prosedür tanımladığımdan tek yaptığı bu prosedürü CALL komutu ile çalıştırmak aslında.

Tanımladığım prosedürler: CREATE_LIST, PRINT_LIST, ADD_TERM_TO_LIST

Eğer 4 girilirse de doğrudan .exe uygulamamız kapansın diye main prosedürünün RET komutundan öncesine koyduğum label'a JMP ile atlıyor.

Main prosedürün yapısı psödö kod ile bir bakıma şöyle:

```
if(AX == 1) createList();  
else if(AX ==2) printList();  
else if(AX == 3) addTermToList();  
else if(AX == 4) System.exit(0);
```

PRINT_LIST PROSEDÜRÜ:

İlk yazdığım prosedür buydu, öncelikle test edebilmek için data segment'te elimle birkaç değer tanımladım:

```
ARR 30, 20, 40, 50, DW 46 DUP(?)
```

```
LINKS DW 2, 0, 3, -1, 46 DUP(?)
```

```
N DW 4
```

```
MIN_INDEX DW 1
```

Bu ödevde linkedlist'i şöyle düşünüyorum:

Index	0	1	2	3
Array	30	20	40	50
Links	2	0	3	-1

Burada ARR dediğim dizi doğrudan raw data'ı girilme sırası ile birlikte içinde tutan bir dizi.

LINKS dizisi ise ARR'deki her terimin hangi terimi gösterdiğini belirten bir index. LINKS dizisini kullanarak PRINT_LIST fonksiyonunu yazacağım aslında. link * 2 bana her zaman offset'i vereceğinden dolayı SHL komutunu da kullanmam gerekecek. Ayrıca son terim dışında her terimi yazdıktan sonra ok simgesini (string tanımlı en yukarıda) yazarak daha da güzel bir görselleştirme yapıyorum.

Prosedürü okursanız göreceksiniz ki en yukarıda ARR dizisi hiçbir sort işlemine tabi tutulmadan doğrudan yazılıyor. ARR dizisinden sonra da aynı işlemi LINKS dizisine yapıyorum. Bu iki dizi print edildikten sonra artık link'leri kullanarak gerçek linkedlist'i (sorted halini) yazdırıyorum. Burada işime yarayacak bir değişken daha var: MIN_INDEX. Bu değişken adından da anlaşılacağı gibi array'in minimum teriminin indeksini tutuyor. Gelecekte bunun yanına MAX_INDEX de ekleyeceğim ve bu index değerlerini yeni bir terim eklerken otomatik olarak güncelleyeceğim. Şimdilik yalnızca MIN_INDEX yeterli çünkü kullanacağım algoritma en küçük terimden başlayıp sırayla LINKS[offset] değerini alıp bir sonraki terime geçerek o terimi yazdıracak.

ARR'nin yazdırılması:

```
; Veriler: kisminin yazdirilmasi
MOV AX, OFFSET DATAS_STR
CALL PUT_STR

MOV CX, N
MOV BX, 0
LB11:
MOV AX, ARR[BX]
CALL PUTN

CMP CX, 1
JZ NO_PRINT
MOV AX, OFFSET COMMA_STR
CALL PUT_STR
NO_PRINT:

ADD BX, 2
DEC CX
CMP CX, 0
JZ END_LB1
JMP LB11
END_LB1:
```

LINKS'in yazdırılması:

```
; Linkler: kisminin yazdirilmasi
MOV AX, OFFSET LINKS_STR
CALL PUT_STR

MOV CX, N
MOV BX, 0
LB22:
MOV AX, LINKS[BX]
CALL PUTN

CMP CX, 1
JZ NO_PRINT2
MOV AX, OFFSET COMMA_STR
CALL PUT_STR
NO_PRINT2:

ADD BX, 2
DEC CX
CMP CX, 0
JZ END_LB2
JMP LB22
END_LB2:
```

ARR ve LINKS kullanarak linkedlist'in sorted bir şekilde yazdırılması:

```
; Listenin okunup linklere gore sorted sekilde yazdirilmesi
MOV AX, OFFSET VALUES_STR
CALL PUT_STR

MOV SI, MIN_INDEX
SHL SI, 1          ; min_index dedigimiz dogrudan indeks o yuzden offset elde etmek icin x2 yap

MOV CX, N
MY_LABEL:
MOV AX, ARR[SI]
CALL PUTN

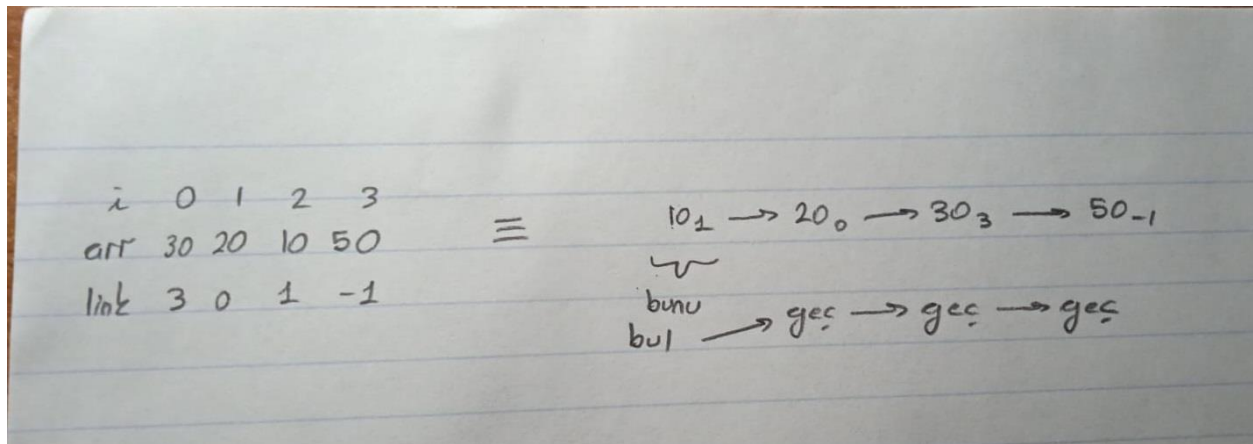
; en son terimden sonra bitirmeden once ok yazilmamali
CMP CX, 1
JZ END_PROC

MOV AX, OFFSET ARROW_STR
CALL PUT_STR

MOV SI, LINKS[SI]
SHL SI, 1          ; linkler dogrudan indeks o yuzden offset elde etmek icin x2 yap

END_PROC:
LOOP MY_LABEL
```

Burada bir bakıma şöyle bir traversal yapıyor:



ADD_TERM_TO_LIST PROSEDÜRÜ:

Bütün ödev aslında bu prosedürü yazmakla ilgili. Ödevin başında kurduğum bir algoritma vardı, psödö kod ile açıklamak gerekirse şöyle yazabilirim:

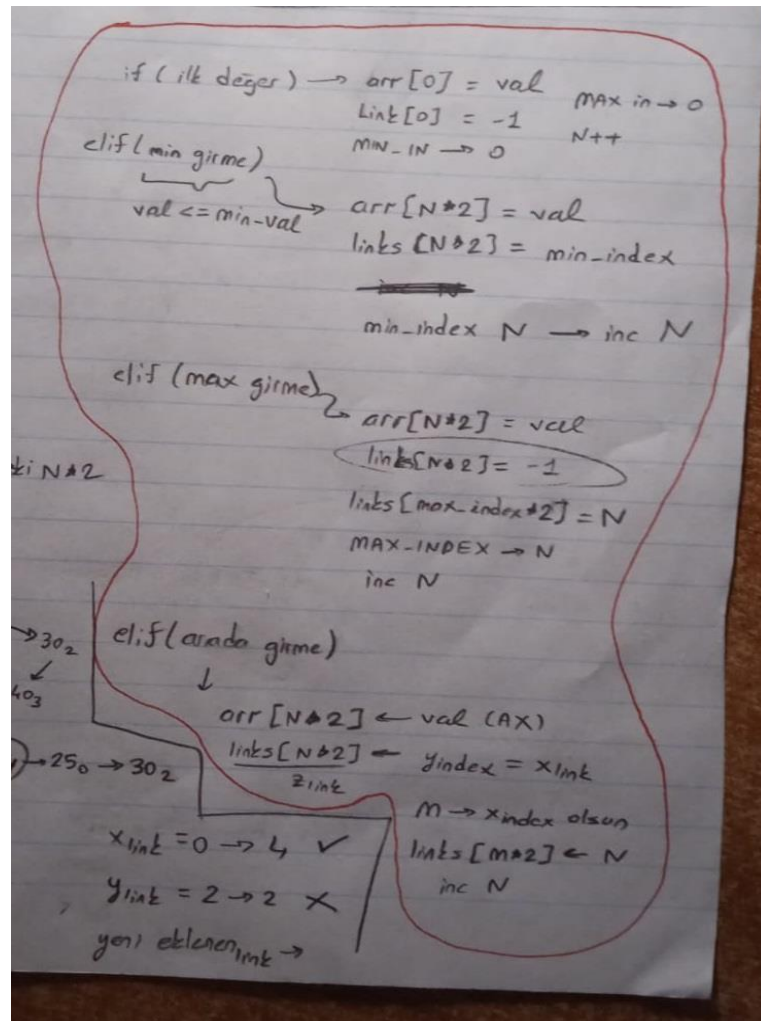
if(ilk terim girildi) ilk terimi ekleme kodu

else if(min terim girildi) min terimi ekleme kodu

else if(max terim girildi) max terimi ekleme kodu

else kesinlikle iki terim arasında terim eklendi öyle ona göre ekle

Bu algoritmayı kurarken bir sürü kağıt üzerinde örnek yaptım ve koşulları belirledim:



İlk değer girme koşulları:

- `ARR[0] = AX` olmalı
- `LINKS[0] = -1` olmalı
- `MIN_INDEX` ve `MAX_INDEX` de 0 olmalı
- İşlemler bittikten sonra `N` arttırılmalı

Koda dönecek olursak:

```
CMP N, 0
JNZ CONT1
; N = 0 yani hiç değer yok ilk değer girilecek
MOV ARR[0], AX
MOV LINKS[0], -1
MOV MIN_INDEX, 0
MOV MAX_INDEX, 0
INC N
JMP END_FUNC
CONT1:
```

Buradan sonra kodun geri kalanında kullanacağım için `MIN_VALUE` ve `MAX_VALUE` değerlerini çıkartıp bir değişkende tutuyorum. Aslında bunu yapmama gerek yoktu ancak bir sürü `PUSH`, `POP` yaparak kafamı karıştırmak istemediğim için değişken tanımlamayı tercih ettim:

```
; min deger cikartma
MOV BX, MIN_INDEX
SHL BX, 1
MOV BX, ARR[BX]
MOV MIN_VAL, BX
; max deger cikartma
MOV BX, MAX_INDEX
SHL BX, 1
MOV BX, ARR[BX]
MOV MAX_VAL, BX
```


Bunları da çıkarttıktan sonra psödö kodda verdiğim if/else'lere devam ediyorum.

Minimum değer girilme durumu:

- $ARR[son\ terim] = AX$ (girilen terimi en sona ekleme)
- $LINKS[yeni\ terim] = eski\ MIN_INDEX$ (minimum ekleme yaptığımız için yeni eklediğimizin linki eski minimum olacak sanki arkadan eklemiştir gibi olacak kuyruk kısmına)
- $MIN_INDEX = yeni_eklediğimiz_index$ (minimum eklediğimiz için yeni MIN_INDEX 'imiz dizinin uzunluğu olacak)

Koda dönecek olursak:

```
; min deger girildi
MOV BX, N
SHL BX, 1                ; offset'i elde et

MOV ARR[BX], AX          ; yeni deger en sona eklenecek

MOV DX, MIN_INDEX
MOV LINKS[BX], DX        ; eski min index'i son eklenenin index'i olacak

MOV DX, N
MOV MIN_INDEX, DX        ; son eklenen terim yeni minimum index'imiz

INC N
JMP END_FUNC
```

Maksimum değer girilme durumu:

- yine en sona AX 'i ekle
- $LINKS[en_son_eklenen] = -1$ (yeni max eklediğimiz için -1 olacak)
- Eski max değer linki yeni eklediğimizin index'i olacak
- MAX_INDEX yine güncellenecek
- N değeri arttırılacak

Koda dönecek olursak:

```
; max deger girildi
MOV BX, N
SHL BX, 1    ; offset

MOV ARR[BX], AX    ; en son eklenen terimi ekle
MOV LINKS[BX], -1  ; en son eklenen max diye -1 linkli olacak

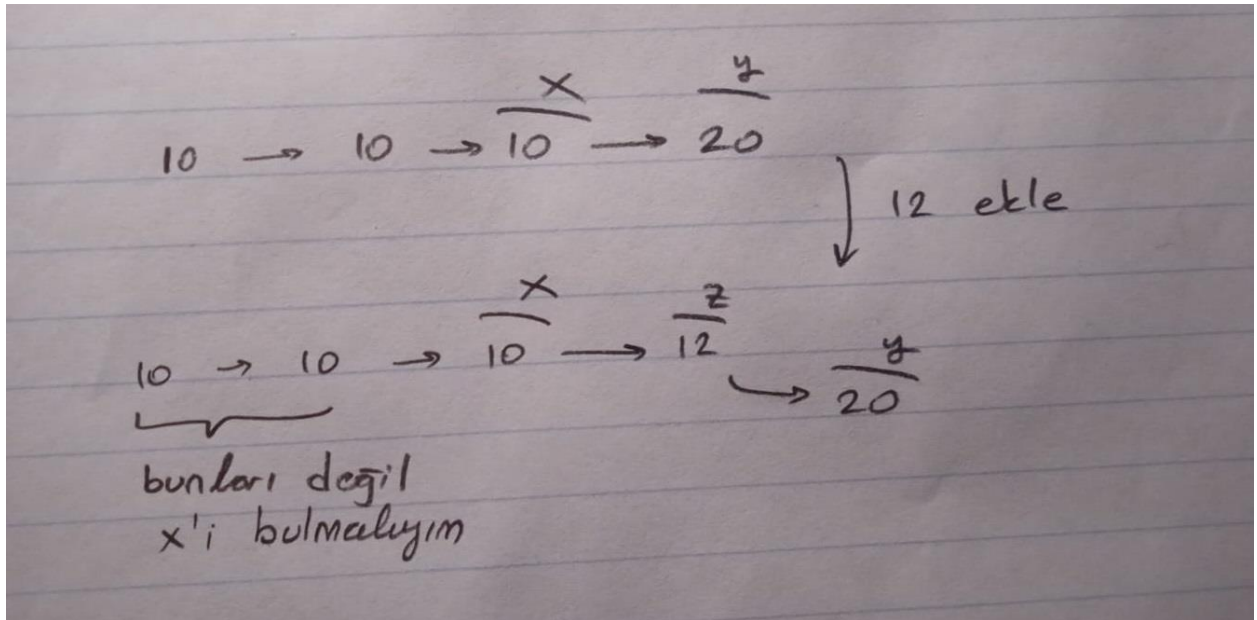
MOV BX, MAX_INDEX
SHL BX, 1
MOV DX, N
MOV LINKS[BX], DX  ; eski max degeri yeni eklenen degere (yeni max'a) bakacak

MOV MAX_INDEX, DX
INC N
```

Arada değır girilme durumu:

Bu belki de en zor durum oluyor çünkü yapmamız gereken şey birbirine eş değırler de olabileceğinden dolayı girilen değırdan önce gelen (kendinden küçük olan EN SON terimi) bulup buna x demek. Sonra da x'ten sonra gelen y'yi bulup x ve y arasına z sokmak olacak.

Görselle açıklamak gerekirse:



Durum böyle olunca bize birkaç değer gerekiyor: x_link ve x_index.

Bunları bir döngü kullanarak belirledim:

x_link belirleme:

```
    ; x_link belirleme
    MOV SI, MIN_INDEX
    SHL SI, 1

    MOV CX, N
MY_LABEL2:
    MOV BX, ARR[SI]

    CMP BX, AX
    JNG STILL_NO ; BX > AX ise kirmali
    JMP END_MY_LOOP
STILL_NO:
    MOV SI, LINKS[SI]
    SHL SI, 1 ; linkler dogrudan indeks o yuzden offset elde etmek icin x2 yap

    LOOP MY_LABEL2
END_MY_LOOP:
    SHR SI, 1

    ; burada SI dedigimiz sey x_link
```

x_link ile x_index belirleme:

```
    ; simdi de SI (x_link) kullanarak x_index bulacagiz
    ; x_index belirleme
    MOV BP, 0

    MOV CX, 0
START_LOOP:
    MOV BX, LINKS[BP]

    CMP BX, SI
    JZ FOUND

    ADD BP, 2

    INC CX
    CMP CX, N
    JNZ START_LOOP
FOUND:

    ; burada CX dedigimiz sey x_index, AX dedigimiz sey yeni_deger, SI dedigimiz sey x_link
```

Şimdi elimde x_index, AX değeri (yeni_deger), x_link olduğundan istediğim işlemleri yapabilirim.

Yapacaklarım sırayla şunlardır:

- 1- en sona yeni terimi ekle
- 2- z teriminin link'i (son eklenen terim) y_index yani x_link olacak
- 3- x teriminin link'i N ile değiştirilecek çünkü yeni eklediğim terim x ve y arasına girecek ve x artık bu terime point edecek
- 4- son olarak da N sayısını arttıracam

Koda dökmek gerekirse:

```
; burada CX dedigimiz sey x_index, AX dedigimiz sey yeni_deger, SI dedigimiz sey x_link
; simdi gerekli seyleri buldugumuza gore istedigimiz islemleri yapabiliriz
; yeni eklenen terimi ekleyelim
MOV BX, N
SHL BX, 1 ; offset
MOV ARR[BX], AX
; yeni eklenen terimin linkini (z_link) gidip y_index = x_link'e esitleyelim
MOV LINKS[BX], SI
; x teriminin indeksini guncelleyelim
MOV BX, CX
SHL BX, 1
MOV BP, N
MOV LINKS[BX], BP
; terim sayisini arttiririm
INC N
```

Bunları da yaptıktan sonra bu prosedürüm bitmiş oluyor.

CREATE_LIST PROSEDÜRÜ:

Bu prosedür de kullanıcıdan bir N değeri alacak ve N kez terim ekleyecek yani N kez ADD_TERM_TO_LIST prosedürünü çalıştıracak.

Koda dökersek şöyle olur:

```

; ----- CREATE_LIST PROSEDUR -----
; ADD_TERM_TO_LIST prosedurunu N kez calistirarak N boyutlu linkedlist olusturur
; PUSH ve POP komutlari kullanmadigimdan AX, CX degisir
;
CREATE_LIST PROC NEAR

    CMP N, 0
    JZ CONT_LB
    MOV AX, OFFSET ALREADY_DEFINED_STR
    CALL PUT_STR

    MOV AX, OFFSET LINE_BREAK
    CALL PUT_STR
    RET
CONT_LB:
    MOV AX, OFFSET AMOUNT_STR
    CALL PUT_STR

    CALL GETN
    MOV CX, AX
    CMP CX, 0
    JLE LOOP_END ; CX <= 0 ise menu kapanmeli

    MOV AX, OFFSET LINE_BREAK
    CALL PUT_STR

```

Bu prosedürü de yazınca ödev bitmiş oluyor. Ek olarak söylemek istediğim birkaç şey daha var:

1- Ödevi pek optimize etmeye çalışmadım daha çok okunabilir olmasına özen gösterdim. Örneğin ADD_TERM_TO_LIST prosedüründe her zaman INC N yaptığımız için bunu bir kez yazıp prosedürün sonuna koyabilirdim ancak böyle yapmadım ve if yapılarını daha okunur bir şekilde belli etsin diye her bir if scope'u içine INC N komutunu koydum.

2- Tüm programda yazmaçların çoğunu hesaplama için kullanan yalnızca ADD_TERM_TO_LIST prosedürü bulunduğundan dolayı çoğu prosedürde PUSH ve POP komutlarını kullanmadım. Zaten gerek de yok hiçbir sorun çıkmıyor.

Yalnızca ADD_TERM_TO_LIST prosedürünün başına ve sonuna CX için PUSH/POP ekledim çünkü eklemezsem CREATE_LIST prosedürü çalışırken döngü değişkeni değiştiğinden N sayısı etkilenmiş oluyor ve prosedür sonsuz bir döngüye girebiliyor ya da doğrudan yanlış bir sayıda terim alıyor.

3- Ödevi başlamadan önce not aldığım algoritmaları da buraya koymak istiyorum, bunların çoğunu değiştirmem gerekti ancak ödevi çat diye yapmış gibi gözükmeyeyim diye arkaplanda neler yaptığımı da göstermek istiyorum:

```
aynı değer girilebilir
```

```
30 20 40 70 50 100 10
```

```
30  
-1
```

```
30 20  
-1 0
```

```
30 20 40  
2 0 -1
```

```
30 20 40 70  
2 0 3 -1
```

```
30 20 40 70 50  
2 0 4 -1 3
```

```
30 20 40 70 50 100  
2 0 4 5 3 -1
```

```
30 20 40 70 50 100 10  
2 0 4 5 3 -1 1
```

```
bunların hepsi if else if else olacak  
önce ilk değer  
sonra min  
sonra arada  
sonra max  
yaz
```

```
İlk değeri ekleme:  
    İlk değer hep -1 ile eklenecek  
  
Minimum ekleme: (<= ile bak)  
    Yeni eklediğin değerin indisi eski minimum değerinin indisi olacak  
  
Max değer ekleme: (>= ile bak)  
    Yeni eklediğin değerin indisi -1 olacak  
    Eski max değerinin indisi şimdi eklediğin terimin indisi yani dizinin uzunluğu -1 olacak  
  
Arada değer ekleme:  
    Eklediğin terimden bir önce (< ile) ve bir sonra (>= ile) gelen terimlerin indislerini bul çünkü işine yarayacak  
    X ve y arasına z giriyor diyelim  
    Y için bir şey yapmayacaksın  
    Z nin linki ynin dizideki indisi olacak  
    X in linki ise yeni eklediğin terimin indisi yani dizilen-1 olacak
```

Programın çıktıları:

4. altmenü ile programı kapatma:

```
C:\>19052025.EXE  
  
Oguzhan Topaloglu 19052025  
Yapmak istediginizi seciniz:  
1 N tane terim girerek listeyi olusturma  
2 Listeyi yazdir  
3 Listeye terim ekle  
4 Programi sonlandir  
  
Secimiz: 4  
  
C:\>
```

1. altmenü ile birden çok terim ekleme:

```
Yapmak istediginizi seciniz:
1 N tane terim girerek listeyi olusturma
2 Listeyi yazdir
3 Listeye terim ekle
4 Programi sonlandir

Secimiz: 1

Kac terim gireceksiniz: 2

Eklenecek deger: 1

Eklenecek deger: 7

Oguzhan Topaloglu 19052025
Yapmak istediginizi seciniz:
1 N tane terim girerek listeyi olusturma
2 Listeyi yazdir
3 Listeye terim ekle
4 Programi sonlandir

Secimiz: _
```

Ve sonrasında terimleri yazdırma:

```
Oguzhan Topaloglu 19052025
Yapmak istediginizi seciniz:
1 N tane terim girerek listeyi olusturma
2 Listeyi yazdir
3 Listeye terim ekle
4 Programi sonlandir

Secimiz: 2

Veriler: 1 , 7
Linkler: 1 , -1
Diziniz: 1 -> 7

Oguzhan Topaloglu 19052025
Yapmak istediginizi seciniz:
1 N tane terim girerek listeyi olusturma
2 Listeyi yazdir
3 Listeye terim ekle
4 Programi sonlandir
```


Dizi tanımlı iken 1. Altmenü çalıştırılmasına izin vermiyorum:

```
Oguzhan Topaloglu 19052025
Yapmak istediginizi seciniz:
1 N tane terim girerek listeyi olusturma
2 Listeyi yazdir
3 Listeye terim ekle
4 Programi sonlandir

Secimiz: 1

Diziniz onceden tanimlandigindan ucuncu menuyu kullanmalisiniz...

Oguzhan Topaloglu 19052025
Yapmak istediginizi seciniz:
1 N tane terim girerek listeyi olusturma
2 Listeyi yazdir
3 Listeye terim ekle
4 Programi sonlandir

Secimiz:
```

3. altmenü ile terimleri ekleme:

```
Oguzhan Topaloglu 19052025
Yapmak istediginizi seciniz:
1 N tane terim girerek listeyi olusturma
2 Listeyi yazdir
3 Listeye terim ekle
4 Programi sonlandir

Secimiz: 3

Eklenecek deger: 50
```

Ve yazdırınca etkisini görme:

```
Oguzhan Topaloglu 19052025
Yapmak istediginizi seciniz:
1 N tane terim girerek listeyi olusturma
2 Listeyi yazdir
3 Listeye terim ekle
4 Programi sonlandir

Secimiz: 2

Veriler: 1 , 7 , 50
Linkler: 1 , 2 , -1
Diziniz: 1 -> 7 -> 50

Oguzhan Topaloglu 19052025
Yapmak istediginizi seciniz:
1 N tane terim girerek listeyi olusturma
2 Listeyi yazdir
3 Listeye terim ekle
4 Programi sonlandir
```

Kafama göre sırayla bir sürü terim ekledikten sonraki durum:

```
Veriler: 1 , 7 , 50 , -10 , 12 , 78 , -45 , -46 , -50 , 45 , 1 , 1 , 1 , 1 , 1 ,
78
Linkler: 10 , 4 , 5 , 0 , 9 , 15 , 3 , 6 , 7 , 2 , 11 , 12 , 13 , 14 , 1 , -1
Diziniz: -50 -> -46 -> -45 -> -10 -> 1 -> 1 -> 1 -> 1 -> 1 -> 1 -> 1 -> 7 -> 12 -> 45
-> 50 -> 78 -> 78

Oguzhan Topaloglu 19052025
Yapmak istediginizi seciniz:
1 N tane terim girerek listeyi olusturma
2 Listeyi yazdir
3 Listeye terim ekle
4 Programi sonlandir
```

En yukarıda raw data, ortada linkler ve en aşağıda da linklerin kullanılarak küçükten büyüğe sıralanmış linkedlist'i görüyoruz.