Öğr.Gör. Furkan ÇAKMAK BLM2021 ALT SEVİYE PROGRAMLAMA

**Son Teslim tarihi:** **02.01.2022**

Ödev 1

Oğuzhan Topaloğlu

Ç19052025 – Grup 3

*Bilgisayar Mühendisliği Bölümü,*

*Elektrik-Elektronik Fakültesi,*

*Yıldız Teknik Üniversitesi*

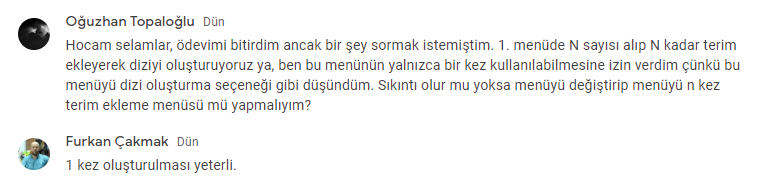
****

Istanbul, 2021

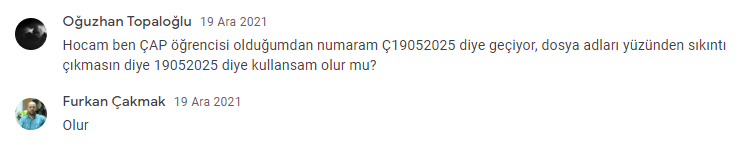
Başlama Öncesi Notlar:

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.



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



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:

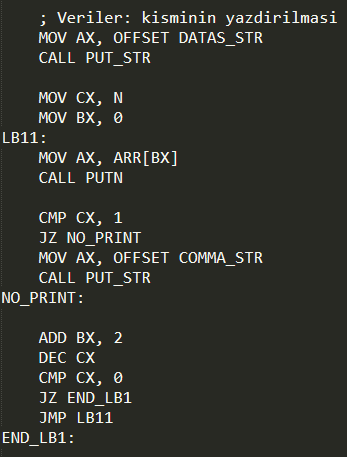
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Index | 0 | 1 | 2 | 3 |
| Array | 30 | 20 | 40 | 50 |
| Links | 2 | 0 | 3 | -1 |

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

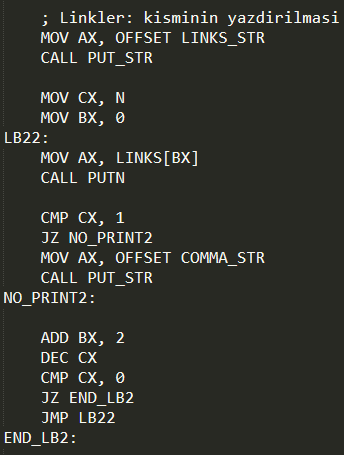
LINKS dizisi ise ARR’deki her terimin hangi terimi gösterdiğini belirten bir index. LINKS dizisini kullanarak PRINT\_LIST fonksiyonunu yazacağım aslında. link \* 2 bana her zaman offset’i vereceğinden dolayı SHL komutunu da kullanmam gerekecek. Ayrıca son terim dışında her terimi yazdıktan sonra ok simgesini (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.

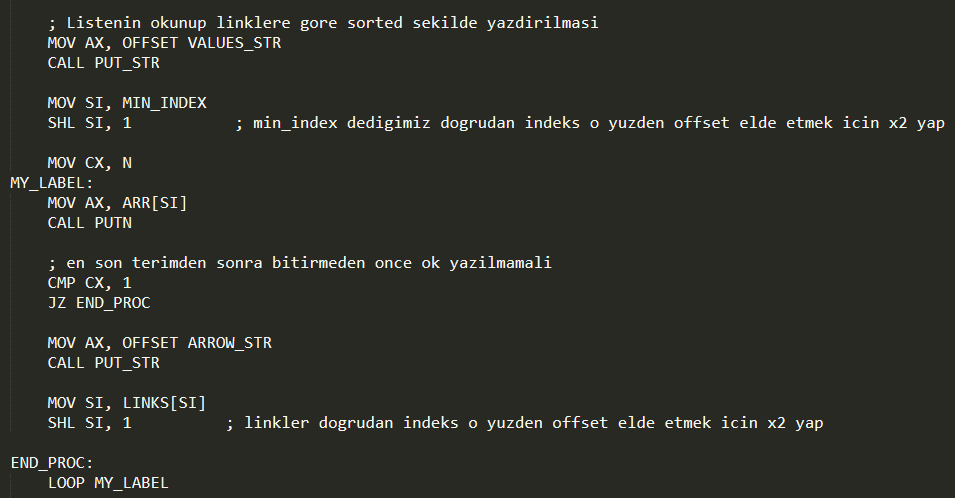
ARR’nin yazdırılması:



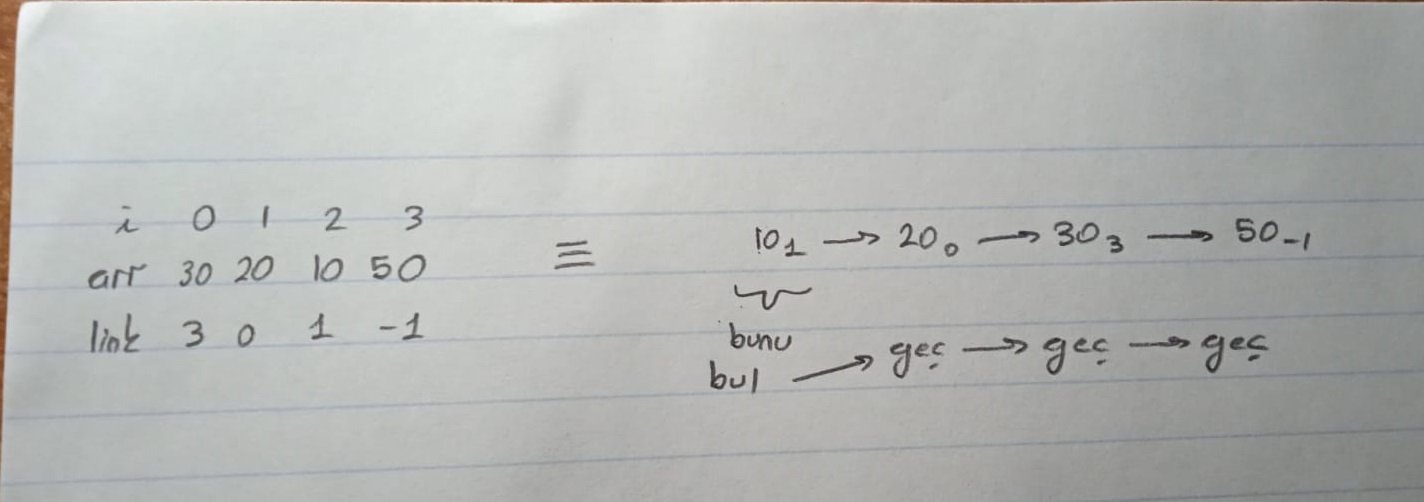
LINKS’in yazdırılması:



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



Burada bir bakıma şöyle bir traversal yapılıyor:



***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:

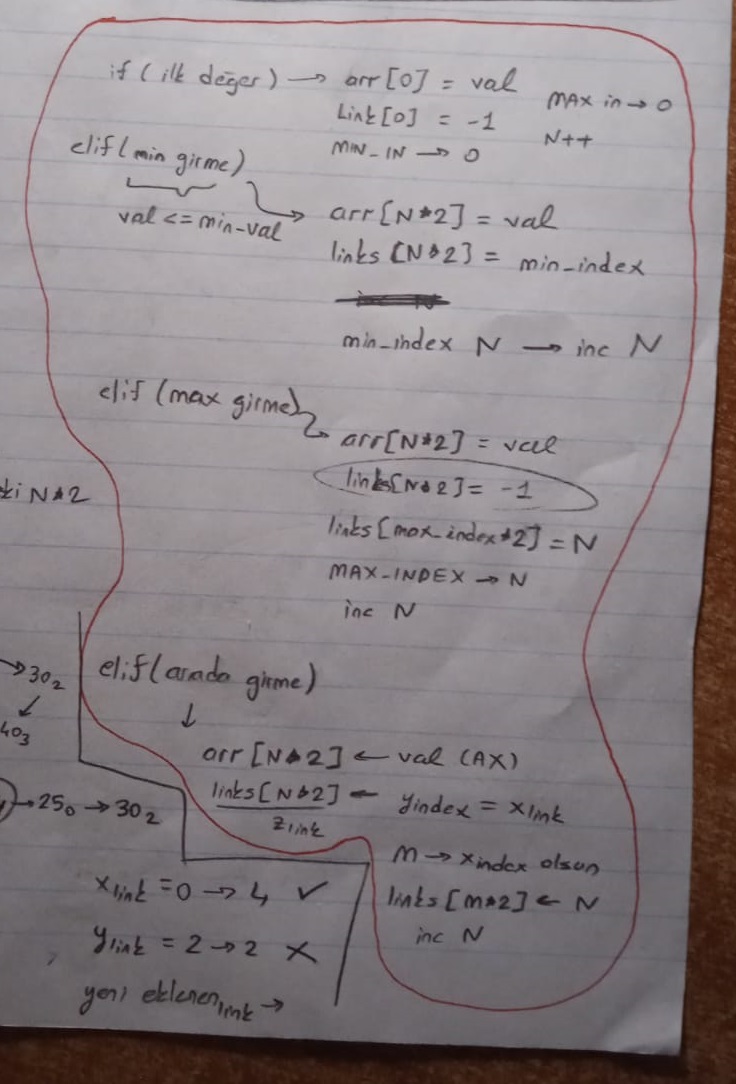
if( ilk terim girildi) ilk terimi ekleme kodu

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

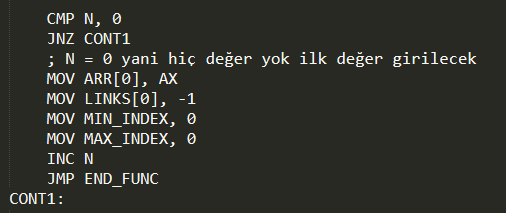
Bu algoritmayı kurarken bir sürü kağıt üzerinde örnek yaptım ve koşulları belirledim:



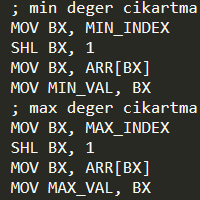
İlk değer girme koşulları:

* ARR[0] = AX olmalı
* LINKS[0] = -1 olmalı
* MIN\_INDEX ve MAX\_INDEX de 0 olmalı
* İşlemler bittikten sonra N arttırılmalı

Koda dökecek olursak:



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:

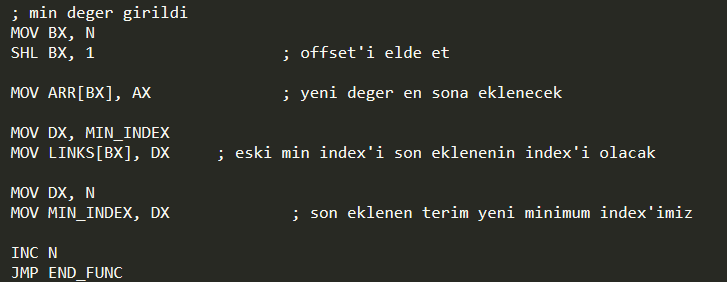


Bunları da çıkarttıktan sonra psödö kodda verdiğim if/else’lere devam ediyorum.

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 eklemişiz gibi olacak kuyruk kısmına)
* MIN\_INDEX = yeni\_eklediğimiz\_index (minimum eklediğimiz için yeni MIN\_INDEX’imiz dizinin uzunluğu olacak)

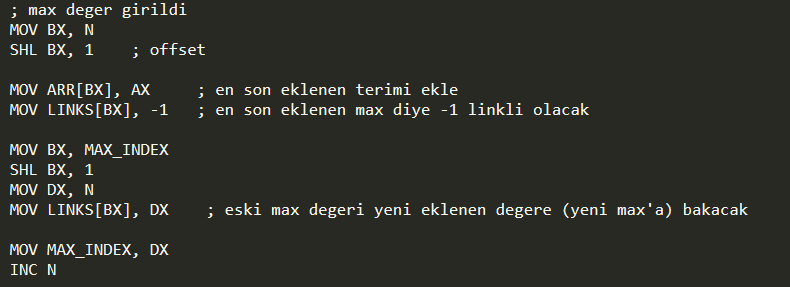
Koda dökecek olursak:



Maksimum değer girilme durumu:  
 - yine en sona AX’i ekle

* LINKS[en\_son\_eklenen] = -1 (yeni max eklediğimiz için -1 olacak)
* Eski max değerin linki yeni eklediğimizin index’i olacak
* MAX\_INDEX yine güncellenecek
* N değeri arttırılacak

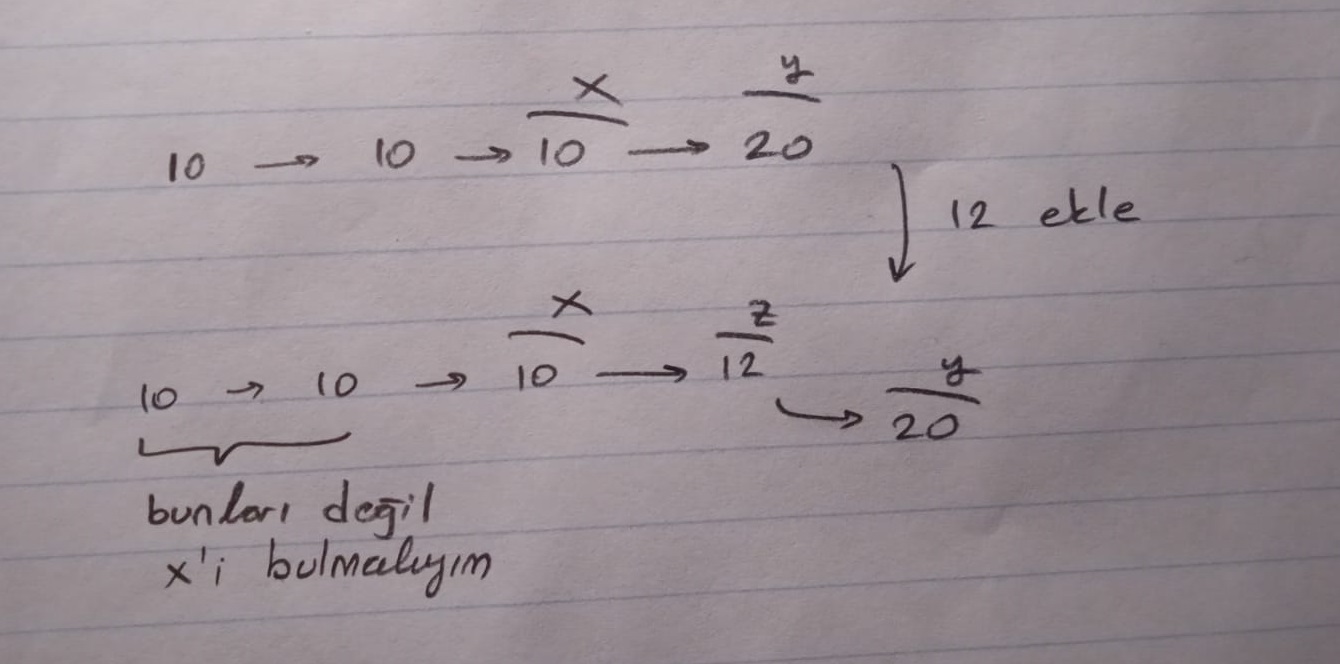
Koda dökecek olursak:



Arada değer girilme durumu:

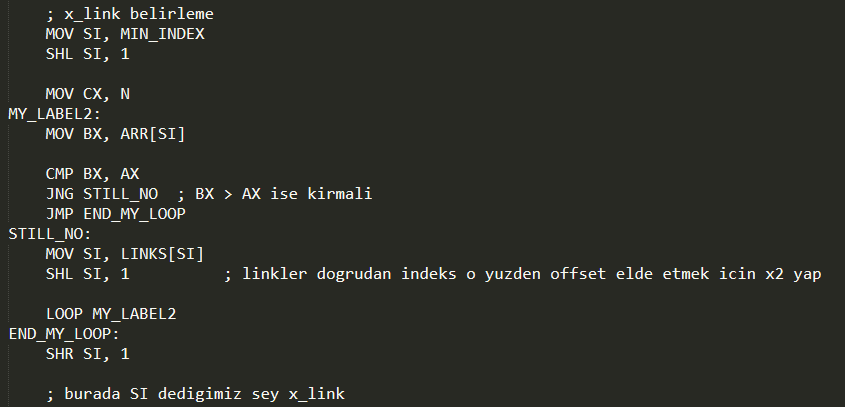
Bu belki de en zor durum oluyor çünkü yapmamız gereken şey birbirine eş değerler de olabileceğinden dolayı girilen değerden önce gelen (kendinden küçük olan EN SON terimi) bulup buna x demek. Sonra da x’ten sonra gelen y’yi bulup x ve y arasına z sokmak olacak.

Görselle açıklamak gerekirse:

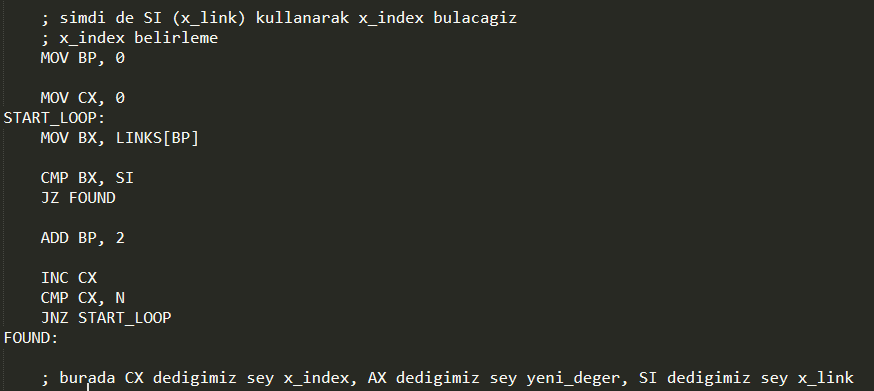


Durum böyle olunca bize birkaç değer gerekiyor: x\_link ve x\_index.

Bunları bir döngü kullanarak belirledim:

x\_link belirleme:  


x\_link ile x\_index belirleme:



Şimdi elimde x\_index, AX değeri (yeni\_deger), x\_link olduğundan istediğim işlemleri yapabilirim.

Yapacaklarım sırayla şunlardır:

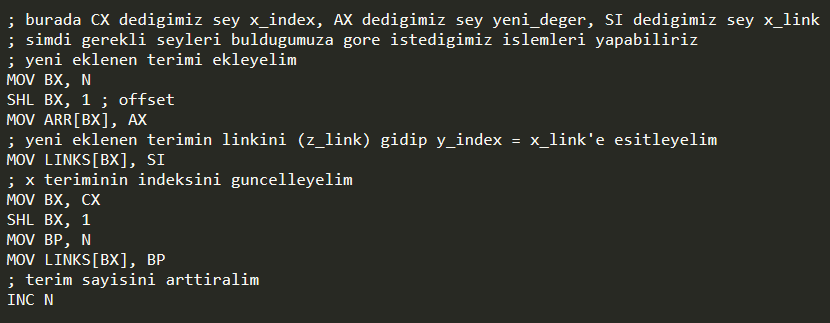
1- en sona yeni terimi ekle

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 arttırırım

Koda dökmek gerekirse:

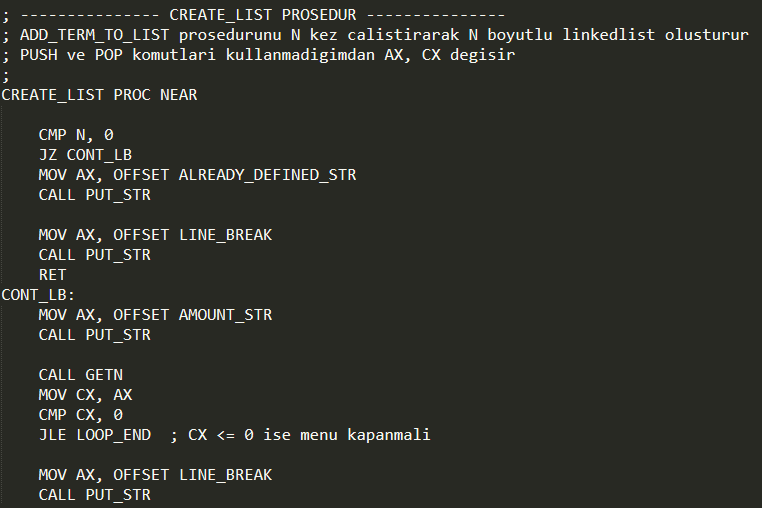


Bunları da yaptıktan sonra bu prosedürüm bitmiş oluyor.

***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:



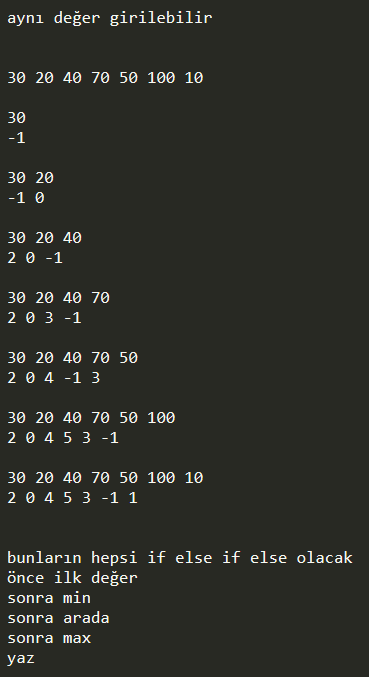
Bu prosedürü de yazınca ödev bitmiş oluyor. Ek olarak söylemek istediğim birkaç şey daha var:

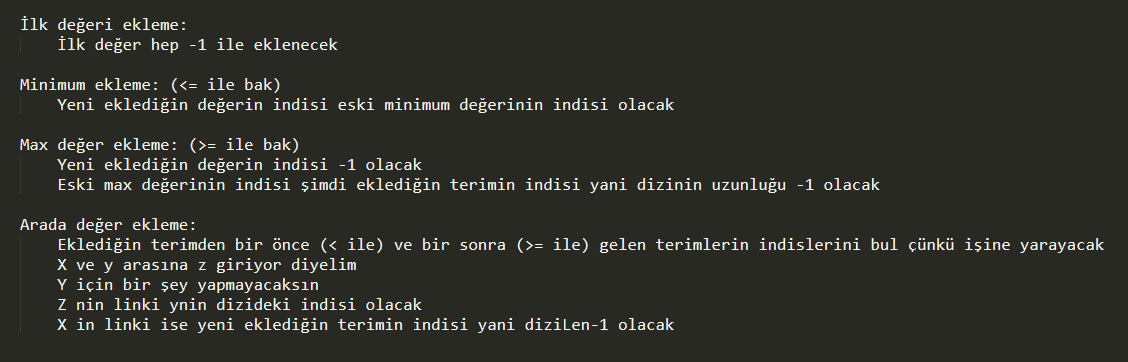
1- Ödevi pek optimize etmeye çalışmadım daha çok okunabilir olmasına özen gösterdim. Örneğin ADD\_TERM\_TO\_LIST prosedüründe her zaman INC N yaptığımız için bunu bir kez yazıp prosedürün sonuna koyabilirdim ancak böyle yapmadım ve if yapılarını daha okunur bir şekilde belli etsin diye her bir if scope’u 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.

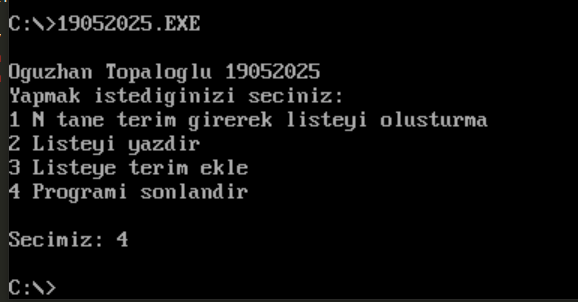
3- Ödeve başlamadan önce not aldığım algoritmaları da buraya koymak istiyorum, bunların çoğunu değiştirmem gerekti ancak ödevi çat diye yapmış gibi gözükmeyeyim diye arkaplanda neler yaptığımı da göstermek istiyorum:

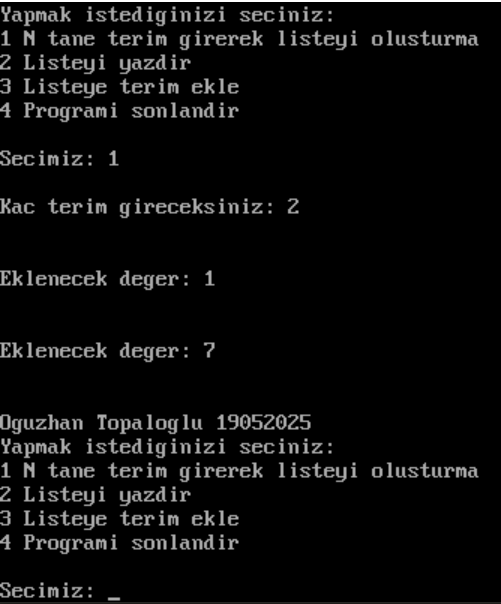


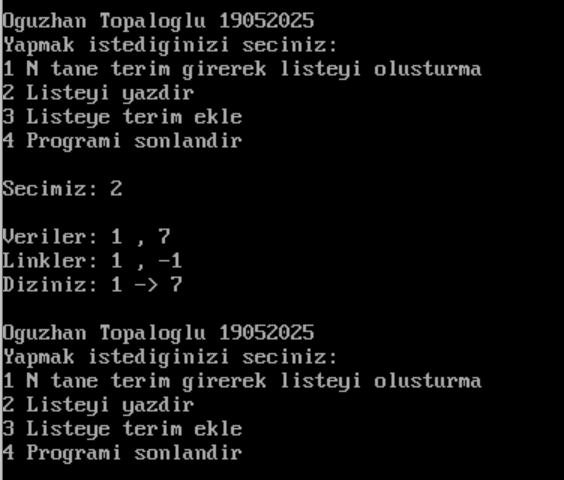


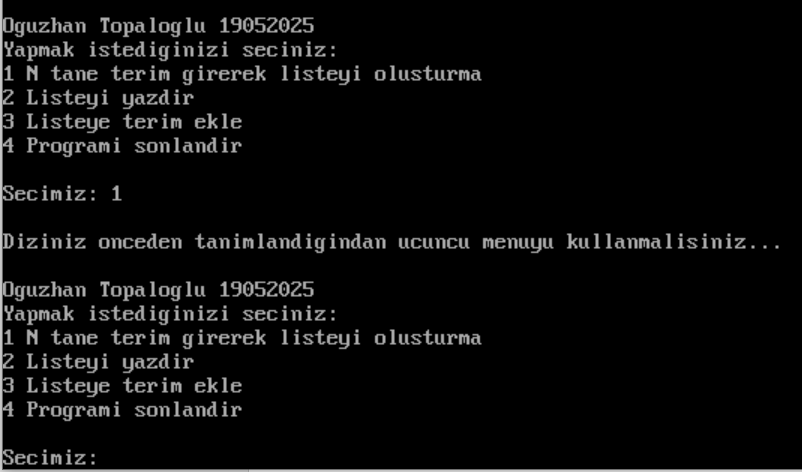
Programın çıktıları:

4. altmenü ile programı kapatma:

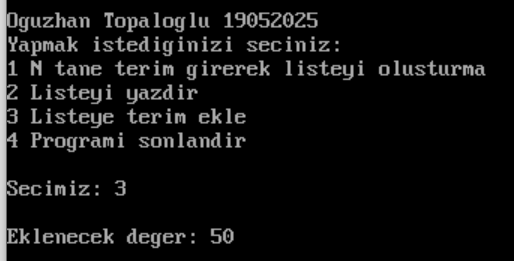


1. altmenü ile birden çok terim ekleme:

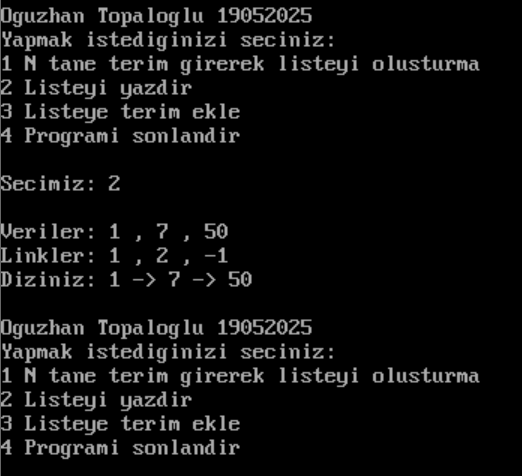
Ve sonrasında terimleri yazdırma:

Dizi tanımlı iken 1. Altmenü çalıştırılmasına izin vermiyorum:

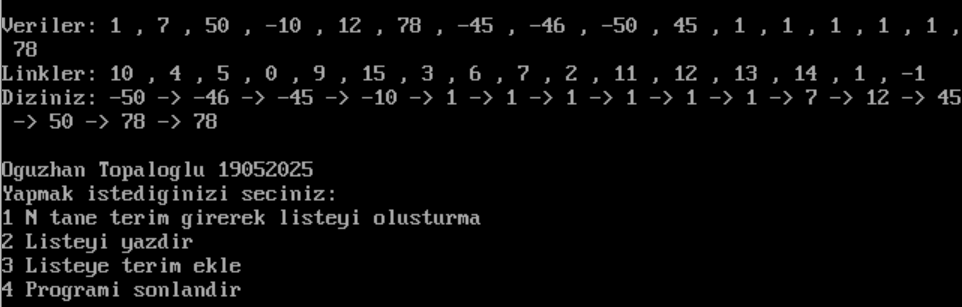
3. altmenü ile terimleri ekleme:



Ve yazdırınca etkisini görme:



Kafama göre sırayla bir sürü terim ekledikten sonraki durum:



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