

**VERİ YAPILARI DERSİNİN İNTERNET ÜZERİNDEN UZAKTAN
EĞİTİMİ**

İsmail PEKGÖZ

**YÜKSEK LİSANS TEZİ
ELEKTRONİK VE BİLGİSAYAR EĞİTİMİ**

**GAZİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

**HAZİRAN 2006
ANKARA**

İsmail PEKGÖZ tarafından hazırlanan VERİ YAPILARI DERSİNİN İNTERNET ÜZERİNDEN UZAKTAN EĞİTİMİ adlı bu tezin Yüksek Lisans tezi olarak uygun olduğunu onaylarım.

Doç.Dr. Abdullah Çavuşoğlu
Tez Yöneticisi

Bu çalışma, jürimiz tarafından Elektronik ve Bilgisayar Eğitimi Anabilim Dalında Yüksek lisans tezi olarak kabul edilmiştir.

Başkan: : Prof.Dr.Ömer Faruk Bay

Üye : Doç.Dr.Abdullah Çavuşoğlu

Üye : Y.Doç.Dr.H.Şakir Bilge

Üye : _____

Üye : _____

Tarih : 23 / 06 / 2006

Bu tez, Gazi Üniversitesi Fen Bilimleri Enstitüsü tez yazım kurallarına uygundur.

TEZ BİLDİRİMİ

Tez içindeki bütün bilgilerin etik davranış ve akademik kurallar çerçevesinde elde edilerek sunulduğunu, ayrıca tez yazım kurallarına uygun olarak hazırlanan bu çalışmada orijinal olmayan her türlü kaynağa eksiksiz atıf yapıldığını bildiririm.

İsmail PEKGÖZ

VERİ YAPILARI DERSİNİN İNTERNET ÜZERİNDEN UZAKTAN EĞİTİMİ

(Yüksek Lisans Tezi)

İsmail PEKGÖZ

**GAZİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

Haziran 2006

ÖZET

İnsanların iletişim, çalışma ve eğitim biçimlerini internet değiştirmiştir. İnternet dünya çapında bilgi, fikir ve ürünlerin el değiştirmesini kolaylaştırırken eğitim ve öğretim alanında da yeni ufuklar açmıştır. Her geçen gün daha fazla üniversite İnternet'e bağlanmakta ve derslerini web üzerinde yayınlanabilmesi imkanı sayesinde akademik imkanları insanlara sunmaktadır. Günümüzde eğitim ve öğretimle uğraşanlar bilgi ve becerilerini klasik derslik dışındaki ortamlarda da öğrencilere taşımaktadırlar. Bu çalışmada amaç Veri Yapıları Dersinin işlenişini her yönüyle ve verimli olacak şekilde internet ortamında uygulamaktır. Dersin amacı, temel veri yapıları (yığın, kuyruk, liste, ağaç , ikili sıralama ağacı, vb.) konularını ve ilgili algoritmaları tanıtmaktır. Ayrıca öğrencilere, problemlere uygun veri yapılarını belirleyebilme ve farklı algoritmaların etkinliğini karşılaştırabilmeleri imkanı sağlamaktır.

Bilim Kodu : 702
Anahtar Kelimeler : Uzaktan eğitim, veri yapıları
Sayfa Adedi : 101
Tez Yöneticisi : Doç. Dr. Abdullah Çavuşoğlu

**DISTANCE EDUCATION OF DATA STRUCTURES LESSON
OVER THE INTERNET**

(M.Sc. Thesis)

İsmail PEKGÖZ

**GAZİ UNIVERSITY
INSTITUTE OF SCIENCE AND TECHNOLOGY**

June 2006

ABSTRACT

Internet has changed the communication, working, education habits of the mankind. It has facilitated worldwide, the exchange of the knowledge, products and thoughts. In addition, it has expedited new fronts in the field of education . Everyday an increasing number of universities are connected to the internet and paving a way to execute the lectures over the internet, enabling academic possibilities to the people. Nowadays, people in the education field are carrying their knowledge and capacities to their students, not only in the classrooms, but in the other mediums as well. In this study, the target has been, carrying the lecture notes of the Data Structures over the internet. The aim of the lesson is providing the basic data structures (lists, stacks, keaps, trees etc.) and relevant algorithms. In addition providing the students, the ability to select appropriate data structures to their problems and also the ability compare the efficiencies of different algorithms.

Science Code :702

Key Words : Distance education,data structures

Page Number: 101

Adviser : Doç Dr. Abdullah Çavuşoğlu

TEŞEKKÜR

Çalışmalarım süresince değerli yardım ve katkılarıyla beni yönlendiren, aynı zamanda Veri Yapıları Ders notlarını kullanmama izin veren hocam, Doç. Dr. Abdullah ÇAVUŞOĞLU'na, yine kıymetli tecrübelerinden faydalandığım hocam, Yrd. Doç. Dr. H. Haldun GÖKTAŞ'a , ayrıca çalışma arkadaşlarım, Murteza KARSLI, Recep KILINÇ ve Ayhan TETİK'e manevi destekleriyle beni hiçbir zaman yalnız bırakmayan çok değerli eşim Hafize PEKGÖZ'e teşekkürü bir borç bilirim.

İÇİNDEKİLER

Sayfa

ÖZET.....	iv
ABSTRACT.....	v
TEŞEKKÜR.....	vi
İÇİNDEKİLER.....	vii
ŞEKİLLERİN LİSTESİ.....	ix
RESİMLERİN LİSTESİ.....	xii
1. GİRİŞ.....	1
2. WEB SİTESİ TASARIMINDA KULLANILAN BAZI PROGRAMLAR.....	5
2.1. HTML(Hypertext Markup Language).....	5
2.2. FrontPage.....	5
2.3. PHP(Personal Home Page).....	7
2.4. MySql.....	8
2.5. AdobePhotoshop.....	9
2.6. JavaScript.....	9
2.7. HTML Dosyalarının İçine PHP Kodlarını Ekleme.....	10
2.8. Fireworks.....	10
2.9. Swish.....	11
2.10. PHP MyAdmin.....	11
3. HAZIRLANAN WEB SİTESİNİN TANITIMI.....	12
3.1. Web Sitesindeki Dersin Tanıtımı.....	13
3.2. Web Sitesindeki Dersin Konularının Tanıtımı.....	14
3.3. Web Sitesindeki Çözümlü Örnekler Bölümü.....	15

Sayfa

3.4. Web Sitesindeki Linkler Bölümünün Tanıtımı.....	16
3.5. Web Sitesindeki Anket Bölümünün Tanıtımı.....	17
3.6. Web Sitesindeki Ziyaretçi Defteri Bölümünün Tanıtımı.....	17
3.7. Web Sitesindeki Dersin Görsel Anlatımı.....	18
4. INTERNET ÜZERİNDEN EĞİTİMİ YAPILAN VERİ YAPILARI DERSİNİN İÇERİĞİ.....	9
4.1. Programlama Prensipleri.....	19
4.2. Bağlı Listeler (Linked Lists).....	19
4.2.1. Tek bağlı doğrusal listeler (One way linked lists).....	20
4.2. 2. Tek bağlı dairesel listeler.....	37
4.2. 3. İki bağlı doğrusal listeler.....	47
4. 2. 4. İki bağlı dairesel listeler.....	58
4.3. Ağaçlar.....	68
5. SONUÇ VE ÖNERİLER	72
KAYNAKLAR.....	75
EKLER	76
EK-1 Web sitesinin yapılmasında kullanılan HTML kodları.....	77
EK-2 Web sitesinin yapılmasında kullanılan PHP kodları.....	90
ÖZGEÇMİŞ.....	101

ŞEKİLLERİN LİSTESİ

Şekil	Sayfa
Şekil 4.1. Tek bağlı doğrusal listelerde node uzayı..	20
Şekil 4.2. Tek bağlı doğrusal listelerde model hafıza alanı	21
Şekil 4.3. İçi doldurulmuş model hafıza alanı	24
Şekil 4.4. Procedure çalıştıktan sonra node uzayı.....	24
Şekil 4.5. Newnode function ı çalışınca elde edilen node uzayı.....	27
Şekil 4.6. Function cuthead ile avail dizisine fonksiyon atılması.....	27
Şekil 4.7. Function last ın çalışması ile oluşan yapı....	28
Şekil 4.8. Birleştirilecek L1 ve L2 listeleri	28
Şekil 4.9. Procedure cocatenate ile birleştirilmiş liste	28
Şekil 4.10. Procedure free ile oluşturulmuş yapı	29
Şekil 4.11. Procedure addhead ile dizinin başına node ekleme	31
Şekil 4.12. Function copy ile list i kopyalama.....	31
Şekil 4.13. Function locate ile list e içinde node bulma	32
Şekil 4.14. Verilen node u function member ile arama	34
Şekil 4.15. Function advance ile node atlama.....	34
Şekil 4.16. Function cuthead ve addhead ile node kesip ekleme	35
Şekil 4.17. Tek bağlı dairesel node uzayı	37
Şekil 4.18. İçi dolu hafıza alanı.....	37
Şekil 4.19. Function cuthead ile node koparma	39
Şekil 4.20. Cuthead ile birleştirilecek L1 ve L2 listeleri	40
Şekil 4.21. Cuthead ile birleştirilmiş liste	40

Şekil	Sayfa
Şekil 4.22 Addhead ile node eklenecek L1 listesi	41
Şekil 4.23. Addhead ile node eklenmiş liste	41
Şekil 4.24. Function copy ile dizi kopyalama	42
Şekil 4.25. Function copy ile tamamlanmış kopyalama	43
Şekil 4.26. Function locate ile node un yerini bulma.....	43
Şekil 4.27. Function member ile node arama.....	44
Şekil 4.28. Function advance ile bir node ilerleme	44
Şekil 4.29. Function delete ile bir node un silinmesi	45
Şekil 4.30. İki bağlı doğrusal listelerde node uzayı	47
Şekil 4.31. İki bağlı doğrusal listelerde boş hafıza alanı.....	47
Şekil 4.32. İki bağlı doğrusal listelerde dolu hafıza alanı	49
Şekil 4.33. Fuction cuthead ile ilk node koparma.....	50
Şekil 4.34. Fuction last ile en son node ulaşma	51
Şekil 4.35. Concatenate ile birleştirilecek L1 ve L2 listeleri	51
Şekil 4.36. Concatenate ile birleştirilmiş liste.....	52
Şekil 4.37. Procedure free ile diziyi avail dizisine ekleme	52
Şekil 4.38. Procedure free ile avail eklenmiş dizi.....	53
Şekil 4.39. Procedure addhead ile diziyi node ekleme	53
Şekil 4.40. Function advance ile bir node ilerleme	56
Şekil 4.41. Function delete ile istenilen node silme	57
Şekil 4.42. İki bağlı dairesel listelerde dolu hafıza alanı.....	59
Şekil 4.43. İki bağlı dairesel listelerde dizinin ilk node unu silme	60

Şekil	Sayfa
Şekil 4.44. İki bağlı dairesel listelerde ilk node silinmiş dizi	61
Şekil 4.45. İki bağlı dairesel listelerde son node bulma.....	61
Şekil 4.46. İki bağlı dairesel listelerde L1 ve L2 listelerini birleştirme	62
Şekil 4.47. İki bağlı dairesel listelerde birleştirilmiş liste.....	63
Şekil 4.48. İki bağlı dairesel listelerde diziye yeni node ekleme	63
Şekil 4.49. İki bağlı dairesel listelerde yeni node eklenmiş dizi.....	64
Şekil 4.50. İki bağlı dairesel listelerde dizi tanımı.....	64
Şekil 4.51. İki bağlı dairesel listelerde cons işlemi.....	65
Şekil 4.52. İki bağlı dairesel listelerde kopyalanacak liste.....	65
Şekil 4.53. İki bağlı dairesel listelerde kopyalama işlemi.....	66
Şekil 4.54. İki bağlı dairesel listelerde data nın yerini bulma işlemi	67
Şekil 4.55.(a) Ağaç yapısı1 (b) Ağaç yapısı2	68
(c) Ağaç yapısı3	69
Şekil 4.56. İki bağlı node ların kullanıldığı ikilik ağaç.....	69
Şekil 4.57. Printree procedure ü yapısı	70

RESİMLERİN LİSTESİ

Resim	Sayfa
Resim 3.1. Web sitesinin internet üzerindeki ana sayfa görünümü	12
Resim 3.2. Web sitesindeki dersin tanıtım.....	13
Resim 3.3. Web sitesindeki dersin konu başlıkları	14
Resim 3.4. Web sitesindeki dersin çözümlü örnekleri.....	15
Resim 3.5. Web sitesindeki linler bölümü	16
Resim 3.6. Web sitesindeki anket bölümü.....	17
Resim 3.7. Web sitesindeki ziyaretçi defteri bölümü.....	17
Resim 3.8. Web sitesindeki görsel anlatım	18
Resim 5.1. Web sitesinin tasarımına ait anket sonucu.....	73
Resim 5.2. Veri yapıları dersi' nin içeriğine ait anket sonucu.....	74

1. GİRİŞ

Bilgi çağı denince akla ilk gelenlerden birisi bilgisayar ve Internet denilen sanal dünyadır. Bilgisayar ve Internet, günümüzde eğitim-öğretim başta olmak üzere iş dünyası, ekonomi, hizmet sektörü ve daha birçok alanda etkin bir şekilde kullanılarak insanlığa büyük yarar sağlamıştır. İnsanlar varolan bilgilerini Internet ortamına aktararak sanal bir dünya da bilgilerini paylaşma imkanı elde etmişlerdir. Şahıs veya kurumlar Internet sayesinde dünyanın her köşesine hızlı bir biçimde ulaşabilmişlerdir. Bilgisayar teknolojisi, 1980'li yıllarda, Internet ise 1990'lardan itibaren yoğun bir şekilde eğitim ve öğretim alanına girmeye başlamıştır. Internet ve bilgisayar teknolojilerindeki hızlı gelişim eğitim-öğretim kurumlarının ve diğer işletmelerin iş yapma biçim ve süreçlerinde faydalı değişiklikler yapmalarına da neden olmuştur.

Bu çalışmada bizim amacımız komple bir dersin işlenişini her yönüyle ve verimli olacak şekilde internet ortamında uygulamaya çalışmaktır. Bu durumda öğrencilerimiz klasik eğitim ve öğretim anlayışından farklı olarak her şeyin sınıflarda ve ders saatinde öğrenilmesinin aksine, internet üzerinden eğitimde, dersi internet'e bağlanabildikleri her yerden ve istedikleri anda takip ederek öğrenebilme şansına sahip olacaklardır. Bu da verilen dersin başarı düzeyini artıracaktır.

Internet üzerinden eğitim ile insanlar okula gitmeden, yollarda zaman harcamadan eğitim-öğretim alabilirler. Okullardan uzak yerlerde yaşayanlar, iş, sağlık ve ailesel nedenlerden dolayı bulundukları ortamı terk edemeyenler eğitim ve öğretimlerini internet üzerinden alabilirler. Doğru yöntem ve teknoloji kullanıldığı, öğrenciler ve öğretmenler arasında iletişim sağlandığı takdirde internet ortamındaki eğitim ve öğretim yüz yüze eğitim kadar başarılı olabilir. Internet ortamındaki eğitim ve öğretim, öğrenci eğitiminde kullanılabileceği gibi, şirket içi eğitimlerde, öğretici ve yer problemi yaşanan durumlarda ve yüz yüze eğitim ve öğretime destek vermek amacıyla da kullanılabilir. Internet ortamındaki eğitim ve öğretimin etkili olması için dikkatli bir planlama, eğitim ve öğretimde verilecek olan ders içeriklerinin anlaşılması ve öğrenci ihtiyaçlarının göz önünde bulundurulması gerekir.

Kullanılacak olan teknoloji ise bu duruma göre belirlenir. İnternet ortamındaki eğitim ve öğretim özürülü insanlarımıza da çok faydalı olabilir. Bu kişiler okula gitme gereksinimi duymadan bulundukları yerden eğitim ve öğretimlerine devam edebilirler [1].

İnternet üzerinden yapılan eğitim ve öğretim aynı zamanda uzaktan eğitim ve öğretim olarak da adlandırılabilir. Uzaktan eğitim ve öğretimin tarihi geçmişine göz attığımızda ilk olarak 1728’de Boston Gazetesinin mektup ile Stenografi Dersleri verdiğini görüyoruz. 1890’da Avustralya Queensland Üniversitesi kampüs dışında açık bir eğitim ve öğretim programı yürütmüştür. 1920’lerde aynı türden bir eğitim metodu Columbia Üniversitesi tarafından gerçekleştirilmiştir. 1930’lara gelindiğinde bazı okulların radyoyu kullanarak eğitim ve öğretim verdiğini görüyoruz. 1950’lerde Amerika Birleşik Devletleri’de askeri amaçlı olarak uzaktan eğitim programlarını uygulamaya koymuştur [1].

Ülkemizde ilk uzaktan eğitim ve öğretim uygulamaları geniş çapta mektupla öğretim ile başlamıştır. Bu kapsamda özellikle yabancı dil eğitim ve öğretimleri göze çarpmaktadır. Ülkemizdeki uzaktan eğitim ve öğretim uygulamalarına baktığımızda televizyon tabanlı uygulamalar ağırlık kazanmaktadır. Anadolu Üniversitesi, Milli Eğitim Bakanlığı’na bağlı Açık öğretim Lisesi ve Açık İlköğretim Okulu uygulamaları buna en çarpıcı örneklerdir. İnternet tabanlı uygulamalara baktığımızda, Orta Doğu Teknik Üniversitesi’nde 1998 yılında başlayan İnternet’e Dayalı Asenkron Eğitim ile, tamamen İnternet ortamında “Bilgi Teknolojileri Sertifika Programı” başlatıldığını görüyoruz [1].

Yapılan bu çalışmada internet üzerinden eğitim uygulaması için web tabanlı bir model oluşturulmuştur. Bu modelde web tabanlı bir eğitim-öğretim sitesi tasarlanıp hazırlanmıştır. Hazırlanan web sitesinde Veri Yapıları Dersi nin detaylı bir şekilde yayınlanması hedeflenmiştir. Web sitesinin hazırlanması ve dersin yayınlanmasına kadar geçen süreçte neler yapıldığını kısaca şöyle sıralayabiliriz:

Amaç ve hedefin tespit edilmesi: Bu web sitesini hazırlamaya başlamadan önce amacımızın ne olduğunu ve bu sitenin kimlere yönelik olacağını belirlemeye çalıştık. Amaç, Veri yapıları dersi'ni kapsamlı ve detaylı olarak internet üzerinden yayınlamaktır. Hedef kitlemiz ise öncelikli olarak bilgisayar bölümlerinde eğitim ve öğretim gören öğrenciler olacaktır. Ayrıca bu konulara merak duyan diğer internet kullanıcıları da hedef kitemizi oluşturabilirler.

Araştırma: Amaç ve hitap edilecek hedef kitle belirlendikten sonra, bu konuda bugüne kadar yapılmış çalışmalar titizlikle incelenmiştir. Web tasarımı, çoğunlukla görselliği ön plana çıkaran bir kavram olarak düşünülür. Oysa ki, aslında web sitesi yapmak, tasarım, yapı, teknoloji ve içerik olarak bir web sitesinin yayına hazır hale getirilmesini ifade etmektedir.

İşbirliği: Web sitesinin hazırlanması aşamasında, bu işte uzman kişilerle ve bilgisayar bölümleri ile temasa geçilerek, bilgi alış verişinde bulunmaya özen gösterilmiştir.

Yardımcı materyallerin ve ders içeriğinin hazırlanması: İnternet ortamına aktardığımız Veri yapıları dersinin içeriğine ve hedef kitleye bağlı olarak, web sitesi için metin, ses, görüntü, arama motoru, yardımcı linkler, vb. materyaller kullanılmıştır. Özellikle ders konuları ve konulara uygun çözümlü örnekler bu alanda uzman kişilerle birlikte hazırlanmıştır.

HTML ve PHP kodlarının hazırlanması : HTML, PHP ve JavaScript kodlarını belli bir algorithmaya bağlı olarak birlikte kullanmaya çalıştık. Index, amaç ve tanıtım sayfalarında HTML kodlarını, anket ve ziyaretçi defteri sayfalarında PHP kodlarını, tarih gibi uygulamalarda ise JavaScript kodlarını tercih ettik.

Web sitesini takip eden öğrencilerle iletişim kurulması: Hazırlanan web sitesinde sayfalara, öğrencilere konular bazında seçim imkanı sağlayan linkler konulmuştur. Ayrıca öğrencilerle iletişim sağlamak amacıyla web sitesine hem ziyaretçi defteri hem de duyurular bölümü eklenmiştir.

Web sitesinin test edilmesi ve alan adı altında yayımı: Web sitesi yapımı tamamlandıktan sonra birkaç günlük test yayımı yaparak ayrıntılı bir biçimde web sitesini incelemeye çalıştık. Bu aşamadan sonra son hatalar da düzeltilerek ilgili web adresine web sitesi konulmuştur.

Sitenin güncel tutulması: Web tabanlı eğitim sitesi hayata geçirildikten sonra, bu sitenin güncel tutulması konusu çok önemlidir. Bu yüzden hazırladığımız sitenin içeriğini devamlı gözden geçirmemiz ve site ile ilgili gelen mesajları dikkate almamız gerekir. Ayrıca yapılacak periyodik değerlendirmelerle sitenin içeriği devamlı güncellenmelidir.

Kısacası, web sitesi yapmak önce kolay zannedilen bir iş gibi gelebilir. Nitekim bu iş gerçekten çok kısa sürede ve zorlanmadan da yapılabilir. Fakat gerçek şudur ki, yaptığınız bu web sitesiyle ziyaretçilerinizi etkilemeniz çok zor olabilir. Bütün bunların yanında birde yaptığınız web sitesi eğitim içerikli ise ve de hedef kitleniz öğrenciler ise bu iş çok daha zor olabilir.

2. WEB SİTESİ TASARIMINDA KULLANILAN BAZI PROGRAMLAR

2.1. HTML(Hypertext Markup Language)

İnternetin hayatımıza daha çok girmeye başladığı bu günlerde Web sayfası hazırlamak önemli bir konu haline almıştır. Tabii ki web sayfası hazırlamak için bir çok yazılım kullanılmaktadır. Bu yazılımlardan bir tanesi de HTML'dir.

HTML (HyperText Markup Language / Hareketli-Metin İşaretleme Dili) basitçe, web tarayıcıları ile görebileceğimiz, internet dökümanları hazırlamaya yarayan bir çeşit dildir. HTML, programlama dilleri (C, Pascal, Basic,..) gibi bir programlama mantığı taşımadığından öğrenilmesi ve kullanılması gayet kolay bir dildir. HTML dilden ziyade kabaca metinleri ya da verileri biçimlendirmek ve düzenlemek için kullandığımız komutlar dizisi bile diyebiliriz. HTML dili kullanarak web sayfası hazırlamak için özel bir editöre gerek yoktur. Notepad kullanarak sayfalarınızı hazırlayabilirsiniz [2].

HTML dili, metin, resim ve grafikler gibi sayfa öğelerinin web tarayıcıları tarafından nasıl görüntüleneceğini ve tuşa basma veya farenin tıklanması yolu ile köprünün etkinleşmesi gibi web tarayıcılarının kullanıcının hareketlerine nasıl karşılık vereceğini belirlemek üzere etiketler kullanır [2].

Günümüzde kullanılan web tarayıcılarının çoğunluğu, özellikle Microsoft Internet Explorer ve Netscape Navigator, HTML etiketlerini mevcut standartdakilerin ötesinde tanır. FrontPage programı HTML dosyalarını okur ve yazar ayrıca HTML dilinin bilinmesi gerekmez [2].

2.2. Front Page

FrontPage Microsoft firması tarafından piyasaya sunulan bir web tasarım programıdır. FrontPage programı ile istenilen şekil ve görünümde web siteleri tasarlamak oldukça kolaydır [2].

Web içeriğinin yönetiminin kolay olması için FrontPage, web sayfalarına FrontPage web leri denen küçük birimler halinde düzen verir. Sunucudaki veya bir klasör yapısındaki ilk web e kök web denir. Kök web sayısız ek web ler içerebilir. Kök web in içerdiği web lere kullanıcı web leri veya alt web denir.Kök web in yönetimi genellikle oluşturan kişi tarafından yapılır. Kullanıcı yani alt web lerin yönetimi kişilere dağıtılır [2].

Bir kök web in sahip olacağı alt web lerin aşağıdaki ilkelere göre belirlenmesi gerekir;

A.) Web ne kadar büyükse karşı tarafa yani sunucuya aktarım o kadar yavaş olacaktır. İçerik yönetimi ve eşitleme sorunlarını en aza indirmek için FrontPage web leri birkaç bin dosyadan fazla olmamalıdır [2].

B.) Aralarında kökler bulunan sayfalar genellikle aynı FrontPage web i içerisinde yer alır [2].

C.) Farklı kişilerce yönetilen sayfalar genellikle farklı web lerde yer alır.Web sunucusunda kök web, FrontPage sunucusu kurulurken oluşturulur. Kök web sunucusunun varolan tüm içeriğinden oluşur ve yöneticisi web sunucusunun yöneticisidir.Web sunucusunun yöneticisi isterse alt web ler oluşturabilir ve bunların yönetimini kişilere verebilir [2].

FrontPage metinleri oluşturmanıza, biçimlendirmenize ve sayfa düzenlerini oluşturmanıza imkan verir; FrontPage'in dışında oluşturduğumuz resimleri ekleyebilir;bağlantılar oluşturabilir;kendi web sayfalarınızı düzenlemenizi ve bir sürükle-bırak ortamı içinde bir web de yer alan çeşitli sayfa görünümlerine bağlantılar oluşturmanızı sağlar. Kısacası FrontPage web tasarımında kullanıcılara çok büyük kolaylıklar sağlamaktadır [2].

2.3. PHP(Personal Home Page)

PHP, ilk kez 1994 yılında, Rasmus Lerdorf tarafından, kendi sitesine erişen ziyaretçileri izlemek ve kişisel bilgilerini internet üzerinden yayınlamak amacı ile geliştirildi. O tarihlerde, günümüzdeki gibi gelişmiş web tasarım yazılımlarının bulunmamasından dolayı ,web sayfası yapmak çok daha zordu [3].

Zaman içinde diğer kullanıcılarında PHP yi kullanmaları neticesinde, PHP nin güçlü bir dil yapısına sahip olduğu ve geliştirilebilir nitelikte olduğu hissedildi. Bu gelişmeler üzerine Rasmus Lerdorf, PHP 2.0 sürümünü oluşturdu [3].

PHP 2.0 önemli bug'lar içeriyordu. Zeev Suraski ve Andi Gutmans adlı iki bilgisayar yazılımcısı, PHP 2.0 parser ı (çözümleyici) içindeki bu ciddi problemleri tesbit ettiler. PHP 3.0 sürümü, Rasmus Lerdorf ile birlikte, Zeev Suraski, Andi Gutmans, Stig Bakken, Shane Caraveo ve Jim Winstead'in çalışmaları ile geliştirildi [3].

Daha sonra Zend kütüphanesi desteğini kullanan PHP 4.0 geliştirildi. Zend kütüphanesi, Zeev Suraski ve Andi Gutmans'ın olağanüstü çalışmaları sonucu geliştirilmişti ve PHP yi çok güçlü hale getirdi. Zend adlı bu yeni çalışma ortamı ile bir PHP scriptinin icra zamanında bir önceki sürüme göre 100 kat azalma olduğu test edilmiştir [3].

PHP nin temel özellikleri:

- 1.) PHP, istemci – sunucu yapısının sunucu tarafında çalışan bir script dilidir. Script, tanım olarak kullanıcının araya girmesi olmaksızın icra edilebilen bir komutlar dizisi anlamındadır [3].
- 2.) PHP, çeşitli platformlarda çalışabilen bir dildir. Diğer pek çok script dilinden farklı olarak, hemen hemen tüm işletim sistemleri üzerinde çalışabilir [3].

3.) PHP, ya bir sunucu modülü olarak yada tek başına çalışabilecek bir CGI scripti olarak konfigüre edilebilir [3].

4.) PHP nin veri tabanlarına erişim fonksiyonları standartlaştırılmamıştır; her veri tabanı, farklı ve diğerleri ile uyumsuz bir API kullanılır [3].

5.) PHP HTML kodu içine gömülebilir ve böylece web sayfalarına yerleştirilebilir [3].

HTML ile veri tabanlarına erişim gerçekleştirilemez; oysa PHP ile, veri tabanları ve dosyalarla etkileşim mümkün olduğu gibi, ayrıca e-mail yönetimi ve daha HTML nin yapamadığı pek çok şey gerçekleştirilebilir [3].

2.4. MySql

MySQL, bir veri tabanı yönetim sistemidir. Bunun yanında MySQL aynı zamanda bir ilişkisel veri tabanı yönetim sistemidir.Çünkü, veri tabanının temsilde temel veri modeli olarak ilişkisel model benimsenmiştir. Bu modelde, bilindiği gibi, veri tabanının veri bileşeni, bir tablolar kümesi içine yerleştirilmiştir. Veriyi farklı tablolar içine yerleştirmek bir esneklik sağlar. Gerekli zaman farklı tablolar arasında, içerdikleri aynı veri türündeki alanlar yardımı ile ilişki kurulabilir [3].

MySQL içinde, veri tabanı yönetim sisteminin veri tabanına erişim ve sorgulama işlemleri için kullandığı bileşen ise standart SQL dir [3].

MySQL, aynı zamanda bir açık kaynak yazılımdır.Açık kaynağın anlamı, MySQL i herhangi bir kişinin serbestçe kullanabilmesi yanında,MySQL yapısında değişiklikler de yapabilmesidir [3].

MySQL in ilk planda öne çıkan özelliklerini, çok hızlı işlem yapabilmesi, güvenilir ve kararlı olması ve kolay kullanılabilir olması olarak sıralayabiliriz [3].

MySQL, çok zengin ve faydalı bir fonksiyonlar kümesine sahiptir. Bağlantı kalitesi, hız ve güvenlik faktörleri düşünüldüğünde, MySQL, internet üzerindeki veri tabanlarına erişim bakımından çok uygun bir ürün olarak görülmektedir. Ayrıca popüler tüm programlama dillerinide desteklemektedir [3].

2.5. AdobePhotoshop

Bilgisayarla grafik tasarım artık iyiden iyiye yaşantımıza girmiştir.Dolayısıyla gerek basında, gerek reklam sektöründe bilgisayarsız tasarım yok denecek kadar azdır.Tabii ki tasarımın en önemli unsurlarından biriside resim programlarıdır. Bilgisayarda resim denildiği zaman ilk akla gelen ve en popüler resim işleme programı Adobe Photoshop dur [4].

Profesyonel bir resim işleme programı olan Adobe Photoshop'un kullanım alanının çok geniş olduğu bir gerçektir. Özellikle Web sayfalarının tasarımındaki rolü etkisi çok büyüktür Ancak burada hemen bir hatırlatma yapmamız gerekiyor Photoshop'un çoklu dil (Multi Language) desteği olmadığı için yazılarımızda Türkçe karakterleri ı,ö,İ,ç,ş,ğ kullanamıyoruz. Bu nedenle ya Türkçe karakter setli fontları kullanmamız gerekir ki bunların bir çoğunu internet'den temin edebiliriz [4].

2.6. JavaScript

JavaScript, C dilinde yazılmış programdır. JavaScript Web sayfalarında kullanılabilen yeni bir scripttir. JavaScript in en önemli özelliği HTML sayfası içerisine yerleştirilebilmesidir. JavaScript HTML'nin bir parçasıdır ve içinde bulundurduğu HTML ile birlikte web browseri tarafından yorumlanır. JavaScript ile değişik elemanlar kullanarak HTML sayfaları çok daha iyi düzenlenebilir. Dolayısı ile Javascript programı yazmak demek , bir Web sayfası hazırlamak ve bu sayfadaki HTML kodlarının arasına JavaScript kodları gömmek demektir. JavaScript programlarını yazabilmek için fazla birşeye ihtiyacınız yoktur. Bunun için gerekli olanlardan birincisi iyi bir bilgisayar(Windows 95 ve üstü yüklü vs..) ikincisi ise her bilgisayarda rahatlıkla bulunabilecek bir kelime işlem programıdır. Ayrıca bir zamanlar sadece CGI ile mümkün olan bazı olaylar JavaScript ile de yapılabilir. Yani

JavaScript'in yardımıyla gerçekten de kaliteli Web sayfaları oluşturulabilir. Şu anda Internet üzerinde JavaScript ile ilgili birçok örnek bulunmaktadır [5].

2.7. HTML Dosyalarının İçine PHP Kodlarını Ekleme

HTML dosyasının artık .php uzantılıymış gibi davranmasını ve içindeki kodların PHP motoru tarafından algılanmasını istiyorsak bunu sağlamak için bulunduğumuz web dizini (klasör) içinde bir htaccess dosyası oluşturmak gerekir. Bunu HTML dosyalarına php gibi davranılmasını istediğimiz her klasörde oluşturmamız gerekir. Bu .htaccess dosyasının içine parantez içerisindeki ifadeyi yazarsak (addtype application/x-httpd-php php3 html), artık HTML sayfalarımıza normal PHP dosyalarında olduğu gibi kod parçaları yazabiliriz.

2.8 Fireworks

Fireworks, hem web tasarımında hem de grafik tasarımında kullanılabilen bir programdır. Aynı anda hem pixel hem de vektörel bir tabanda çalışabilme özelliğine sahiptir [6].

Web tasarımında Fireworks ile neler yapabiliriz:

- 1.) Web sayfalarında kullanılmak üzere harika butonlar yapılabilir [6].
- 2.) Fireworks ile göze hoş gelen hareketli butonlar da yapılabilir [6].
- 3.) Herhangi bir nesnenin kolayca dış yüzeyinin boyanmasını sağlar [6].
- 4.) Bir gafiğin, şeklin veya text in renklendirilmesinde kullanılır [6].
- 5.) Fireworks'ün getirmiş olduğu en büyük yeniliklerden biri de, programı kullanarak web sayfalarınız için, çok kolay bir şekilde Pop-Up menüsü hazırlayabilirsiniz [6].

2.9. Swish

Swish, web tasarımında kullanılan önemli yazılımlardan birisidir. Flash vektörel canlandırmalarda ilk ve en gelişmiş örnek olmaya devam etse de yazı konusunda Swish çok daha iyidir. Dosya boyutu oldukça küçük olan Swish yazı hazırlamada kullanıcılara çok yardımcı olmaktadır. Swish de yaptığımız çalışmalar SWI uzantılı olarak kaydedilir. Ayrıca Swish de SWF ve HTML olarak kayıt yapabilme seçenekleride vardır.

2.10. PHP MyAdmin

PHP MyAdmin web tabanlı arayüzü ile özellikle birden fazla kullanıcıli ortamlarda MySQL veritabanını kullanmayı çok kolaylaştırmaktadır. Bu yüzden bir çok hosting şirketi PHP MyAdmin'i hosting hesaplarıyla birlikte kullanıcılara sunmaktadır. Son günlerde bu yazılım o kadar tutulmuştur ki aynı kod örnek alınarak PostgreSQL, Oracle gibi diğer veritabanlarına da uyarlanmıştır. Bu durum PHP MyAdmin i piyasada daha yaygın kullanılır hale getirmiştir.

3. HAZIRLANAN WEB SİTESİNİN TANITIMI

Web sitesinde eğitimi verilen ders Veri Yapıları Dersi'dir. Veri Yapıları Dersi'nin internet üzerindeki ana sayfa görünümü aşağıdaki resimde gösterilmiştir.



Resim 3.1. Web sitesinin internet üzerindeki ana sayfa görünümü

Web sitesinin, ana sayfası sade ve kullanılması kolay uygulamalardan oluşmuştur. Sitenin en başında eğitimini vermeye çalıştığımız Veri Yapıları Dersi'nin ismi yer almaktadır. Sitenin sol tarafına sitenin diğer sayfalarına atlamakta kullandığımız butonları yerleştirdik. Sitenin sağ tarafında ise görüntülü anlatım, duyurular ve arama motoru uygulamalarına yer verdik. Sitenin ortasında ise seçilen uygulamaya ait açıklamanın yer almasını sağladık. Ayrıca sitemize hareketli yazılar ve tarih bilgilerini gösteren bölümler ilave ettik.

Şimdi web sitemizi bölümler halinde detaylı bir biçimde incelersek:

3.1. Web Sitesindeki Dersin Tanıtı

Bu bölümde dersin içeriği hakkında kısa bir tanıtım yapılmıştır.



Resim 3.2. Web sitesindeki dersin tanıtım

3.2. Web Sitesindeki Dersin Konularının Tanıtımı

Veri Yapılarına Dersine ait tüm konu başlıkları bu bölümde yer almaktadır. Derse ait herhangi bir konu başlığı tıklatıldığında zaman o konuya ait içeriği gösteren web sayfası açılmaktadır.



Resim 3.3. Web sitesindeki dersin konu başlıkları

3.3. Web Sitesindeki Dersin Çözümlü Örnekler Bölümü

Veri Yapıları Dersinin çözümlü örneklerine bu bölümden geçilmektedir. Bu bölümde yer alan sorular ve çözümleri alanına tıkladığımız zaman derse ait çözümlü soruların bulunduğu web sayfası açılmaktadır.



Resim 3.4. Web sitesindeki dersin çözümlü örnekleri

3.4. Web Sitesindeki Linkler Bölümünün Tanıtımı

Web sitesinden diğer önemli sitelere link kurmak bu bölümden yapılmaktadır. Seçilen linke göre ona bağlı web sitesine geçiş sağlanır.



Resim 3.5. Web sitesindeki linler bölümü

3.5. Web Sitesindeki Anket Bölümünün Tanıtımı

Web sitesi ile ilgili internet kullanıcılarının düşüncelerini öğrenmek için bir anket uygulaması gerçekleştirilmiştir.

Resim 3.6. Web sitesindeki anket bölümü

3.6. Web Sitesindeki Ziyaretçi Defteri Bölümünün Tanıtımı

Resim 3.7. Web sitesindeki ziyaretçi defteri bölümü

3.7. Web Sitesindeki Dersin Görsel Anlatımı

Bu bölümde derse ait konuları görsel olarak izleme ve dinleme imkanı verilmiştir. Web sitesini ziyaret eden kullanıcılar, internet destekli konu anlatımı alanını tıkladıkları zaman ders görsel olarak da anlatılmaktadır.



Resim 3.8. Web sitesindeki görsel anlatım

4. INTERNET ÜZERİNDEN EĞİTİMİ YAPILAN VERİ YAPILARI DERSİNİN İÇERİĞİ

4.1. Programlama Prensipleri

Büyük bilgisayar programları yazarken karşılaştığımız en büyük sorun programın hedeflerini tayin etmek değildir. Hatta, programı geliştirme aşamasında hangi yöntemleri kullanacağımız hususu da altından kalkılamayacak bir problem değildir.

Bu dersin ana amacı programlama yöntemlerini etkin bir şekilde ortaya koyarak kullanıcılarının boyutları gereğinden büyük olmayan, hızlı çalışabilen, mantıksal yaklaşımları üzerinde toplayan unsurları programcıya iyi bir dil ile aktarmaktır.

Bilgisayar ile ilgili işlerde en başta aklımıza bellek gelir. Bilgisayar programları gerek kendi kodlarını saklamak için veya gerekse kullandıkları verileri saklamak için genelde belleği saklama ortamı olarak seçerler. Belleği etkin şekilde kullanmak için veri yapılarından yararlanmak gerekmektedir. Bunlara örnek olarak karşılaştırma, arama, yeni veri ekleme, silme gibi işlemleri verebiliriz ki bunlar ağırlıklı olarak bellekte gerçekleştirilirler. Aslında bellekte her değişken için yer ayrılmıştır. Örneğin Pascal dilinde tanımladığımız bir x değişkeni için bellekte bir yer ayrılmıştır. Bu x değişkeninin adresini tutan başka bir değişken de olabilir. Buna pointer denir. Pointer içinde bir değişkenin adresini tutar.

4.2. Bağlı Listeler (Linked List)

Tek Bağlı listeler:

- Tek Bağlı Doğrusal listeler
- Tek Bağlı Dairesel listeler

İki Bağlı listeler:

- İki Bağlı Doğrusal listeler
- İki Bağlı Dairesel listeler

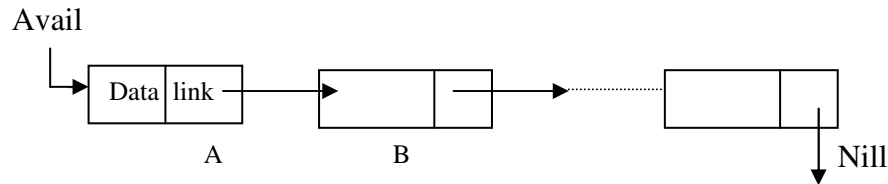
4.2.1. Tek bağılı doğrusal listeler (One way linked list)

Tek bağılı doğrusal dizi, öğelerinin arasındaki ilişki (Logical Connection)'ye göre bir sonraki öğenin bellekte yerleştiği yerin (Memory Location) bir gösterge ile gösterildiği yapıdır.

Bilgisayar belleği doğrusaldır. Bilgiler sıra sıra hücrelere saklanır. Her bir bilgiye daha kolay ulaşmak için bunlara numara verilir ve her birine “node” adı verilir.

Data alanı, numarası verilen node'da tutulacak bilgiyi ifade eder. Link alanı ise bir node'dan sonra hangi node gelecekse o node'un numarası tutulur.

- Link alanı integer türünden bir değer alır.
- Data alanı bölgesindeki numaraları temsil eder.
- Avail: kullanıma hazır, boşta bulunan nodların topluluğunu ifade eder. Avail 1 denilirse bu ifade 1 numaralı ndeyi işaret etmektedir.



Şekil 4.1. Tek bağılı doğrusal listelerde node uzayı

Diyelim ki A, B, C, D gibi bir bloğa ihtiyacımız oldu. Pascal'da uygun komutları kullanarak bu verilerin aralarında bağlantı kurabiliriz. Bu iş için (Şekil 4.2.) deki gibi bir Model Hafıza Alanımız olsun. Şekil 4.1'deki gibi bir Node Uzayını tanımlamak için kullanacağımız program unit node_uz dur.

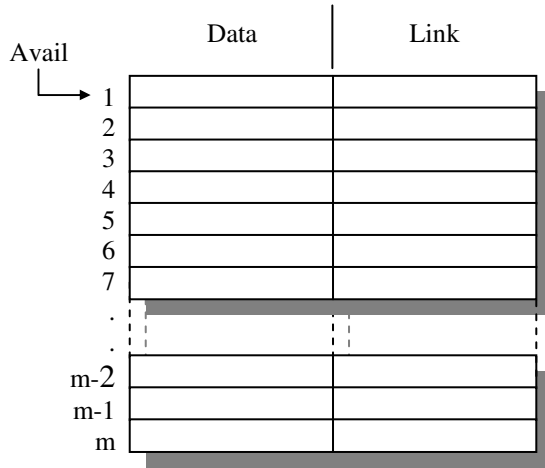
```
Unit node_uz ;
Interface
Const
memory=10; nil=0;
```

```

Var
avail:integer;
node:array [1..memory] of record;
data:real; link:integer;
end;
Implementation
End.

```

Bu altyordamda 10 adet node alanı ayırıyoruz. Avail node uzayında boş kullanılabilir alanları gösterir. Unit Node_uz unitinde avail'i integer, node'u ise record tipli dizi değişken olarak tanımlıyoruz. Node uzayımız 10 adet hafıza alanından oluştuğundan dolayı sabit tanımlama bloğunda memory'yi 10 a eşitledik. Eğer daha büyük hafıza alanı gerekirse sabit tanımlama bloğunda memory'nin değerini arttırabiliriz. Link dediğimiz kısım, bir sonraki node'u gösteren pointer'dır. Integer tipi olması yeterlidir. Data kısmını ise real tipinde bilgi saklayacağımızdan dolayı real olarak tanımladık.



Şekil 4.2. Model hafıza alanı

Bir bağlı listelerde memoryde node,link ve data alanı oluşturulup bir bağlı listelerde node ları başlangıç durumuna getiren program.

```
Unit altyor0
interface (**)
procedure dumpmem;
procedure nodeinit;
implementation (**)
uses Node_uz;
procedure dumpmem;
var
i:integer;
begin
for i:=1 to memory do
writeln ('Node[' ,i:3,']=',node[i].data,',',node[i].link);
end;
procedure nodeinit;
var
i:integer;
begin
avail:=1;
for i:=1 to memory-1 do
node[i].link:=i+1;
node[memory].link:=nill;
end;
end.
```

Fonksiyonlardaki değişkenlerin ömrü fonksiyon sonuna kadar geçerlidir. Fonksiyon bitince hafızada yer kaplamaz. Avail; kullanılmayan node'ları gösterir. Fonksiyon başına VAR yazılmazsa program boyunca yapılan değişiklikler List dizisini etkilemez. Dinamik değişkenler de yer sadece gerektiğinde istenir.

Function ve procedure tanımlama tipleri:

- Call pass by Value
- Call Pass by Reference
- Call Pass by Name
- Call by result

Var

x:integer;

{ Call Pass by Value }

Ekran çıktısı

procedure val(x:integer);

begin

18

x:=x+5;

10

writeln(x);

15

end;

15

10

{ Call Pass by Referance }

20

procedure ref(var x:integer);

5

begin

10

x:=x+5;

writeln(x);end;

{ Call Pass by Name }

procedure name();

begin

x:=x+x;

end;

{ Call by Result }

procedure res();

var x:integer;

begin

x:=x+5;

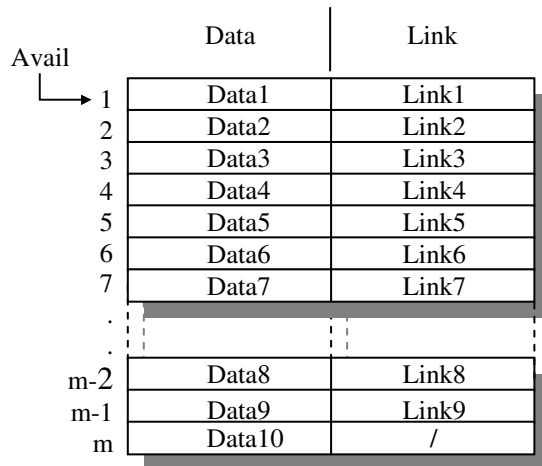
writeln(x);

```

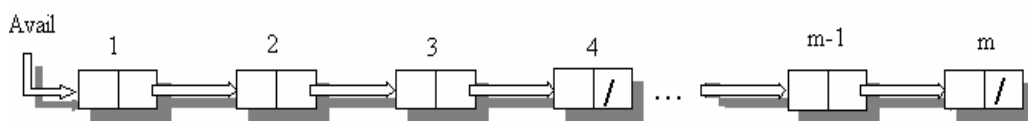
end;
begin
x:=10;
val(13); writeln(x);
ref(x); writeln(x);
name( ); writeln(x);
res( ); writeln(x);
end.

```

Procedure dumpmem, node uzayımızdaki node'ların hem data kısmını hemde linklerini ekrana basıyor. Procedure nodeinit, linklerin birbirine ardarda bağlanmasını sağlar. Burada her link bir sonraki node'u gösterecek şekilde tanımlanıyor. (node[i].link :=i+1). Bir bağlı doğrusal dizi kullanacağımızdan dolayı en son node'un linkini daha önceden Node_uz unitinde 0'a eşitlediğimiz nil'e eşitledik (node.[memory].link :=nil).



Şekil 4.3. İçi doldurulmuş model hafıza alanı



Şekil 4.4. Procedure çalıştıktan sonra node uzayı

UnitAltYor1

Aşağıda yazılan unit ve ilk tanımlanan fonksiyon olan newnode fonksiyonu ile kullanılmak üzere node uzayından bir node koparılır, koparılan bu node avail olarak tanımlıdır kullanılmak için hazır beklemektedir, koparılan bu yeni node'dan sonra node uzayındaki avail diğer hazır bekleyen node'a eşitlenir.

```
unit altyor1;
interface(**)
function newnode:integer;
function cuthead(var list:integer):integer;
function last(list:integer):integer;
procedure concatenate (var L1:integer; L2:integer);
procedure free(list:integer);
implementation(**)
uses node_uz, altyor0;
function newnode :integer ;
var
new:integer;
begin
new:=cuthead(avail);
if new = nil then begin
writeln('nodespace full...') ;
halt;
end;
{else}
newnode:=new;
end;
function cuthead(var {iabile} list:integer):integer;
begin
cuthead:=list;
if list<>nil then list:=node[list].link;
```

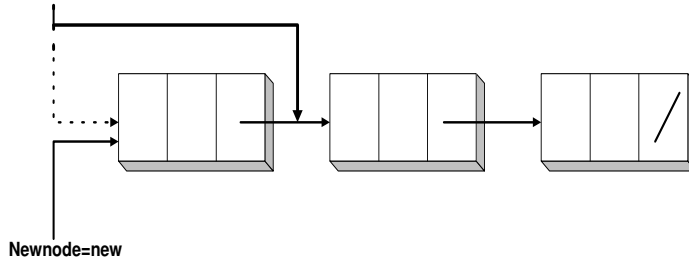
```

end;
function last({the value of}list:integer):integer;(***)
begin
if list<>nill then
while node[list].link<>nill do
list:=node[list].link;
last:=list;
end;
procedure concatenate(var L1:integer;L2:integer);
begin
if L1=nill then L1:=L2 else node[Last(L1)].Link:=L2;
end;
procedure free(list:integer);
begin
concatenate(list,Avail);
Avail:=list;
end;
end.

```

Function newnode : Bellekten gerekli yeri (belli bir değişken için) almak için kullanılır. Bu fonksiyonla node uzayından kullanılmak üzere bir tane node koparılır. Koparılan bu node, node uzayında kullanılmak üzere hazır bekleyen node'dur. Yani avail'dir. Kullanılmak için kullanılan bu yeni node 'dan sonra node uzayındaki avail diğer hazır bekleyen node'a eşitlenir.

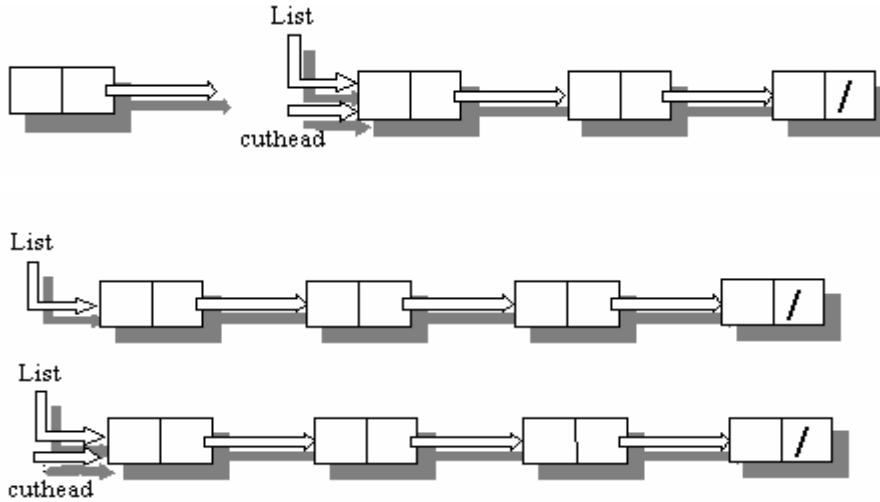
Cuthead fonksiyonu, list isimli bir değişken pas ediliyor (Aslında bu list bir dizi) Cuthead =list diyoruz. Eğer bu list 0 elemanlı bir dizi ise cuthead fonksiyonu geriye 0 döndürecektir. Cuthead ile dizinin ilk node'unu kes gönder diyoruz. Yani; list 0 değilse; 1. node cuthead'in içine alınır. List de ilk node'a eşitken ikinci node'a eşit hale gelir.



Şekil 4.5. Newnodw function 1 çalışınca elde edilen node uzayı

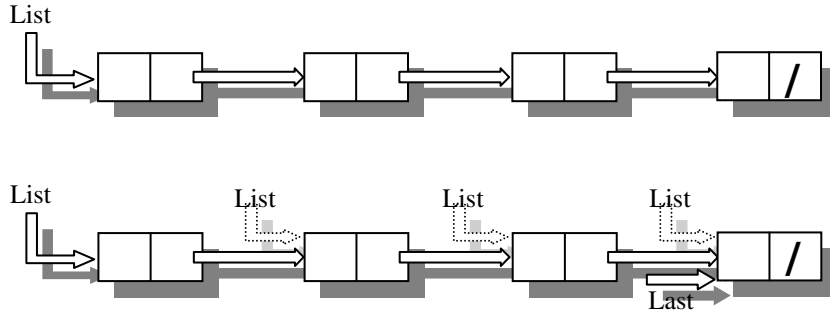
Avail'in 0 olması demek, kullanıma hazır node yok demektir. Bu durumda "Node Space Full" şeklinde bir mesajla karşılarız.

Function cuthead : Biraz önce bahsedilen istenmeyen kutucuğun avail dizisine atılmasını sağlayan fonksiyondur.



Şekil 4.6. Function cuthead ile avail dizisine fonksiyon atılması

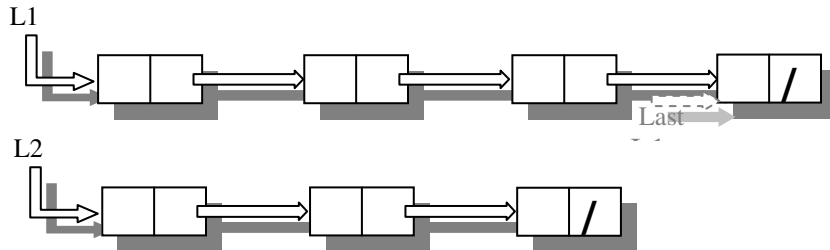
Function last : Bu fonksiyonla sonuncu node bulunur. Öncelikle list diye bir şeyin null olmaması ve linkinin null olmaması şartları kullanılarak sonuncu node bulunur.



Şekil 4.7. Function last ın çalışması ile oluşan yapı

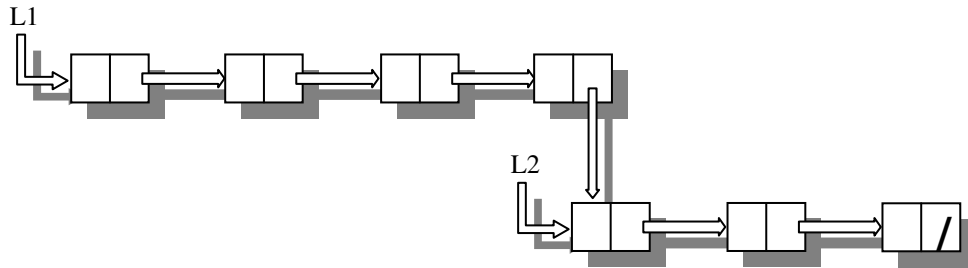
Elimizde bir dizi olsun, bu dizinin en sonuna elemanından sonra bir eleman daha eklememiz istense last fonksiyonunu kullanabiliriz. Eğer dizi sıfır elemanlı ise last'ı yoktur. Last = list olur. Eğer dizi sıfır elemanlı değilse list'i bir ilerletir. list sıfır olana kadar ilerletilir. Sıfır bulununca dizinin sonu bulunmuş demektir.

Procedure concatenate: Diyelim ki elimizde L1 ve L2 listeleri var. Bunları birleştirmek için kullandığımız fonksiyondur.



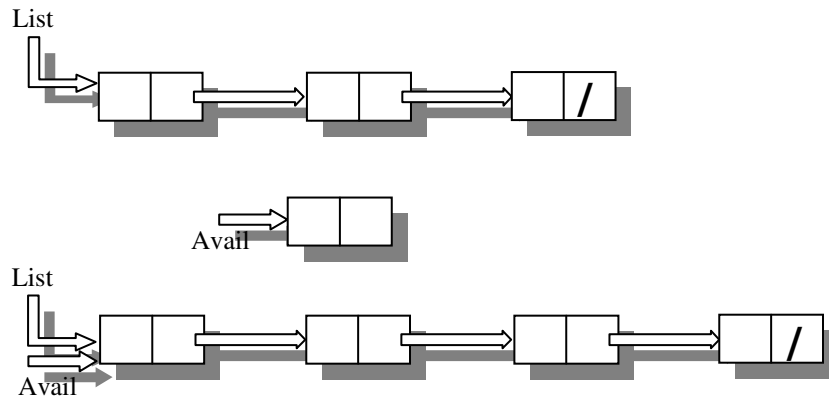
Şekil 4.8. Birleştirilecek L1 ve L2 listeleri

Eğer L1 boşsa L2'yi L1'in arkasına ekliyoruz, L1 dolu ise, L1'in en son node'unu buluyoruz ve daha sonra L1'in last'ı L2 olsun diyoruz.



Şekil 4.9. Procedure concatenate ile birleştirilmiş liste

Procedure free : Elimizde bir liste var. Bu liste ile işimiz bittiğinde bunların bellekte yer işgal etmemesi ve tekrar kullanılabilir hale getirilmesi gerekir. Bu procedure ile kullanılıp işi biten dizi node uzayına gönderilir. Yani list'i avail dizisinin başına ekler ve avail'i list olarak yeniler.



Şekil 4.10. Procedure free ile oluşturulmuş yapı

Concatenate fonksiyonu ile işi biten list dizisini avail dizisinin başına ekliyoruz. Avail=list diyerek işi biten diziyi avail dizisine aktarmış oluyoruz.

Unit altyor2

Unitaltyor2;

interface(**)

procedure addhead(node_:integer; var list:integer);

function cons(data_:real;link_:integer):integer;

function copy(list:integer):integer;

function locate(data_,list:integer):integer;

implementation (**)

uses node_uz,altyor0,altyor1;

procedure addhead(node_:integer; var list:integer);

begin

node_[node_].link:=list;

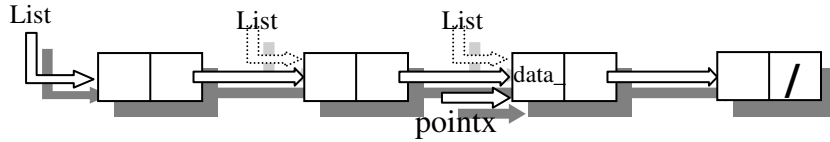
list:=node_;

```

end;
function cons (data_: real; link_: integer): integer;
var
  cons_:integer;
begin
  cons_:newnode;
  cons:=cons_;
  node[cons_].data:=data_;
  node[cons_].link:=link_;
end;
function copy(list:integer):integer;
var
  suret:integer;
begin
  suret :=nill;
  if list<>nill then repeat
    concatenate(suret,cons(node[list].data,nill));
    list:=node[list].link;
  until list=nill;
  copy:=suret;
end;
function locate(data_:real;list:integer):integer;
begin
  locate:=nill;
  while list<>nill do
    if node[list].data<>data_ then list:=node[list].link else begin
      locate:=list; exit;
    end
  end;
end.

```

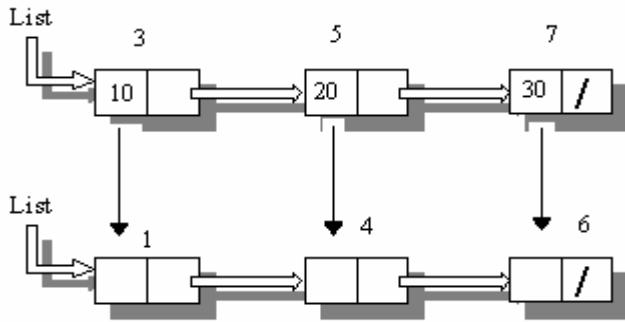
Procedure addhead: Herhangi bir dizinin başına bir node ekler. List dizisi işlem sonunda değişeceğinden var olarak tanımlanmıştır.



Şekil 4.11. Procedure addhead ile dizinin başına node eklenir

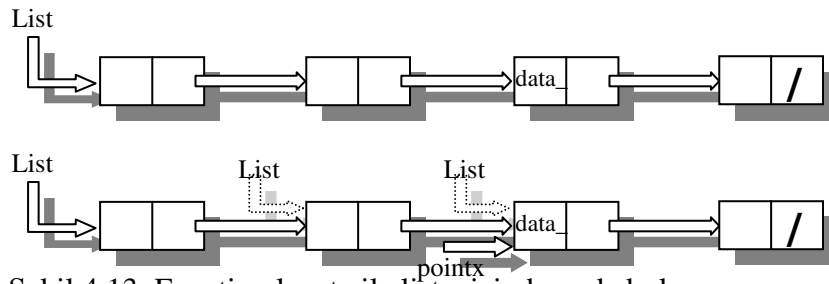
Function cons : Node uzayından alınan node'un data ve link alanını doldurmak için bu fonksiyon kullanılır. Yeni bir node alıp, bu node içine yeni değerler yazma işlemini yapar.

Function copy; Bu function herhangi bir dizinin kopyasını alır listin kopyasını oluşturup geri gönderir. Datalar aynı node nolar farklı olur.



Şekil 4.12. Function copy ile list i kopyalama

Function locate : Bu fonsiyon ile verilen data (data_) 'nın elde bulunan dizide olup olmadığı, eğer varsa data'nın dizinin içerisinde bulunduğu adresi yani numarası bulunur. Liste içinde yer alan bir tane node'u bulmamıza yarar.



Şekil 4.13. Function locate ile list e içinde node bulma

Data_’u datalarla karşılaştırıp aynı olanı ararız. En başta “locate :=nill” ile sıfırlıyoruz. İlk node’un data’sı farklı ise list 2. node’u; o da farklı ise list 3. node’u . gösterebiliriz. Aynı datayı bulunca “locate:=list” yapıp çık.

Unit altyor3

```

interface(**)
function member(node_,list:integer):boolean;
function advance(var point:integer):boolean;
function delete(node_:integer; var list:integer):boolean;
implementation(**)
uses node_uz,altyor0,altyor1,altyor2;
function member(node_,list:integer):boolean;
begin
while (list <> nill) and (list<>node_) do
list:=node[list].link;
member:=(node_=list);
end;
function advance(var point:integer):boolean;
begin
advance:=false;
if point<>nill then
if node[point].link<>nill then
begin

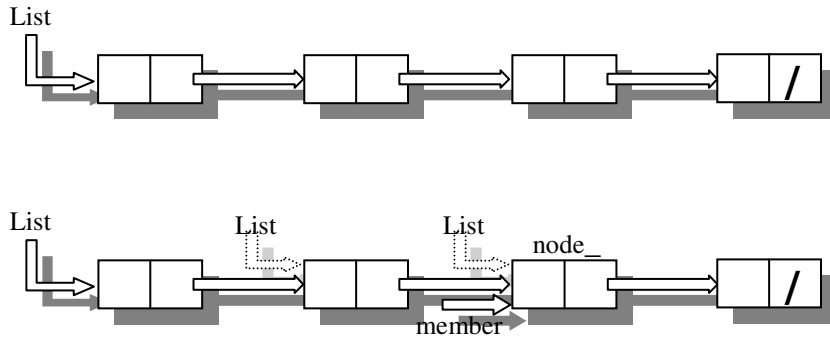
```

```

point:=node[point].link
advance:true;
end;
end;
function delete(node_:integer;var list:integer):boolean;
var
point:integer;
begin
delete:=false;
if list=nil then exit;
if list=node then
begin
addhead(cuthead(list),avail);
delete:=true;
end;
else begin
point:=list;
repeat
if node[point].link=node_ then
begin
addhead(cuthead(node[point].link),avail);
delete:=true;
exit;
end;
until not advance(point);
end;
end;
end.

```

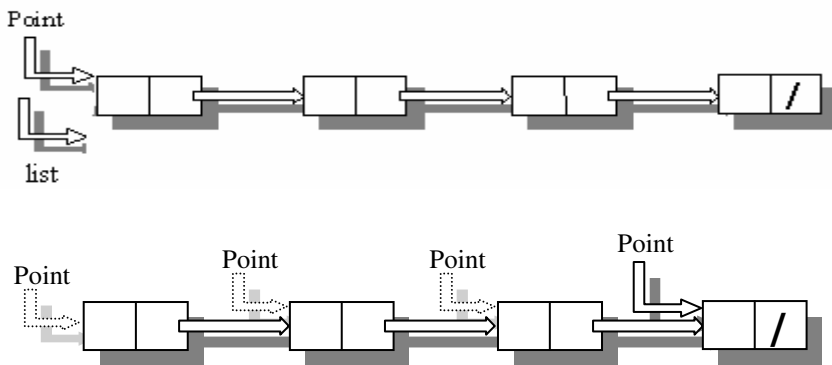
Function member: Verilen node bu listede var mı yok mu? Kontrol eden function'dır. list dizinin başını gösteren elemandır. Node_= list ifadesi karşılaştırma ifadesidir. Node_= list ise member 1, node_<>list ise member 0 'dır.



Şekil 4.14. Verilen node u function member ile arama

Eğer parametre olarak gönderilen node list'te ise mantıksal true, yoksa mantıksal false değerini gönderir.

Function advance: Dizide bir sonraki node'a atlamamızı sağlayan function'dır. Eğer atlama gerçekleşirse mantıksal true, gerçekleşmezse mantıksal false değerini gönderir. Bir dizide elimizdeki pointer'ın gösterdiği node'dan bir sonrakini gösterecek şekilde ilerletir. pointer illa o listenin başını gösterecek diye bir şart yoktur. Eğer point $\neq$ null ise geriye false döner. point dizi üzerinde nerede olduğumuzu gösteriyor. point'in gösterdiği node'un linki

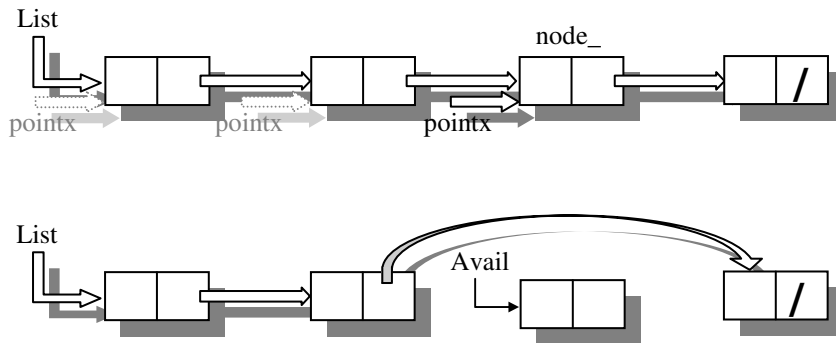


Şekil 4.15. Function advance ile node atlama

Function delete : Dizimizden bir tane node silmek için geliştirilmiş bir function'dır. List: silinmek istenen node'un bulunduğu liste; node_silinmesi istenen node'un numarası.

Eğer dizimiz boş ise hemen function'dan çıkılır ve geriye false değeri döndürür. Cuthead function'u ile listenin ilk node'unu kesip addhead function'u ile avail'in

baş tarafına ekleriz. Delete node_'u listeden silip, avail'in baş tarafına ekler. Eğer silinen liste avail ise, node avail listesinden silinir ve availin başına eklenir. Yani avail o node'dan başlar. Bu işlem advance function'u ile ilerleyemeyinceye kadar devam eder.



Şekil 4.16. Function cuthead ve addhead ile node kesip ekleme

Eğer eleman dizinin herhangi bir yerindeyse, önce arama işlemini yapıyoruz. node_u bulduktan sonra cuthead, node[point].link diyoruz. Cuthead ile o node'u koparıyoruz. Daha sonra addhead ile avail'e ekliyoruz.

Program Nodeini: Yukarıda tanımlanan tüm procedure ve function'lar aşağıdaki gibi bir programda kullanılabilir. Programun uses kısmına daha önceden yazdığımız procedure ve function'ların bulunduğu unit isimlerini ekliyoruz.

Program nodeini;

uses

node_uz,altyor0,altyor1,altyor2,altyor3;

var

i,new:integer;

begin

nodeinit; i:=5;

writeln('delete:',delete(i,avail));

writeln('av=',avail);

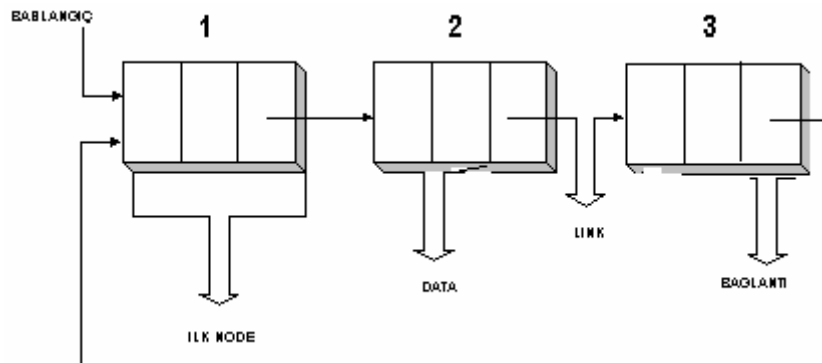
```
writeln('lastav',last(avail));  
writeln('cuth=',cuthhead(avail));  
new:=newnode;  
writeln('new=',new,'cut..',cuthhead(avail),'ava=',avail);  
dumpmem;  
end.
```

Yukarıdaki program çalıştırılırsa aşağıdaki çıktı elde edilir.

```
Delete:TRUE  
Av=5  
Lastav10  
Cuth=5  
New=1cut..2ava=3  
Node[ 1 ]= 2  
Node[ 2 ]= 3  
Node[ 3 ]= 4  
Node[ 4 ]= 5  
Node[ 5 ]= 6  
Node[ 6 ]= 7  
Node[ 7 ]= 8  
Node[ 8 ]= 9  
Node[ 9 ]= 10  
Node[ 10]= 0
```

4.2. 2. Tek bağılı dairesel listeler

Tek bağılı dairesel listelerde; doğrusal listelerdeki bir çok işlem aynı uygulanır; fakat burada dikkat edilmesi gereken, son node'dan sonra null değil list olarak belirlenen ilk node'un gelmesidir.



Şekil 4.17. Tek bağılı dairesel node uzayı

Tek bağılı dairesel listelerde nodeinit, cuthead, last, concatenate, free, addhead, copy, locate ve member function ve procedure'leri değişikliğe uğrar, diğerleri aynı doğrusal listelerdeki gibidir.

Procedure Nodeinit: Bu procedure burada bir bağılı dairesel listelerin node'larını başlangıç durumuna getirir.

Avail	Data	Link
1	Data1	Link1
2	Data2	Link2
3	Data3	Link3
4	Data4	Link4
5	Data5	Link5
6	Data6	Link6
7	Data7	Link7
...	...	...
m-2	Data8	Link8
m-1	Data9	Link9
m	Data10	Link10

Şekil 4.18. İçi dolu hafıza alanı

Procedure nodeinit:

```

var
i:integer;
begin
Avail:=1;
for i:=1 to memory do
node[i].link:=i+1;
node[memory].link:=avail;
end;

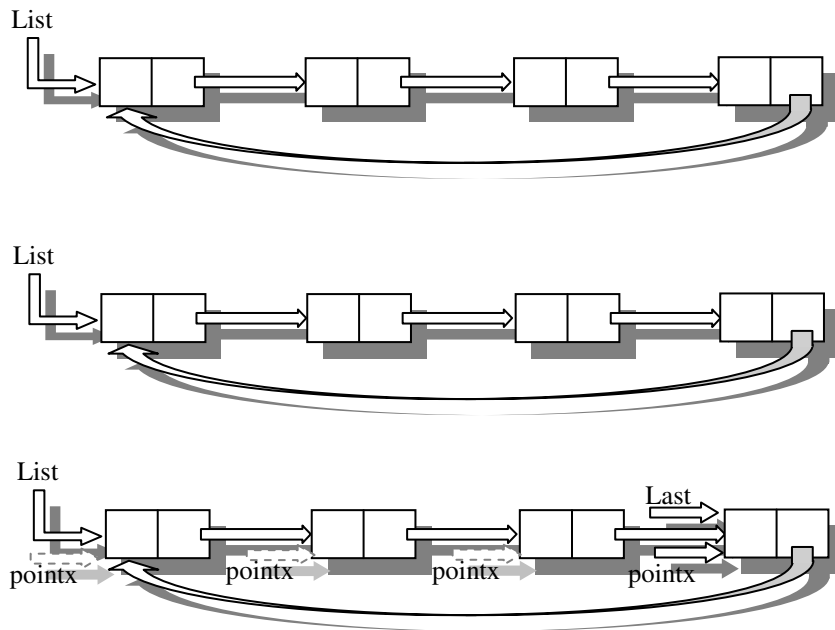
```

Function cuthead: Tek bağlı dairesel listelerde node uzayından ilk node'u koparma.

```

Function cuthead(var list:integer):integer;
var
Pointx:integer;
Begin
Cuthead:=list;
if list<>nill then
if node[list].link=list
then list:=nill
else begin
pointx:=node[list].link
node[last(list)].link:=pointx;
list:=pointx;
end;
end;

```



Şekil 4.19. Function cuthead ile node koparma

Function Last: Tek bağlı dairesel listelerde son node'u bulma

Function last(list: integer): integer;

Var

Point:integer;

Begin

Point:=list;

If point<> nil then

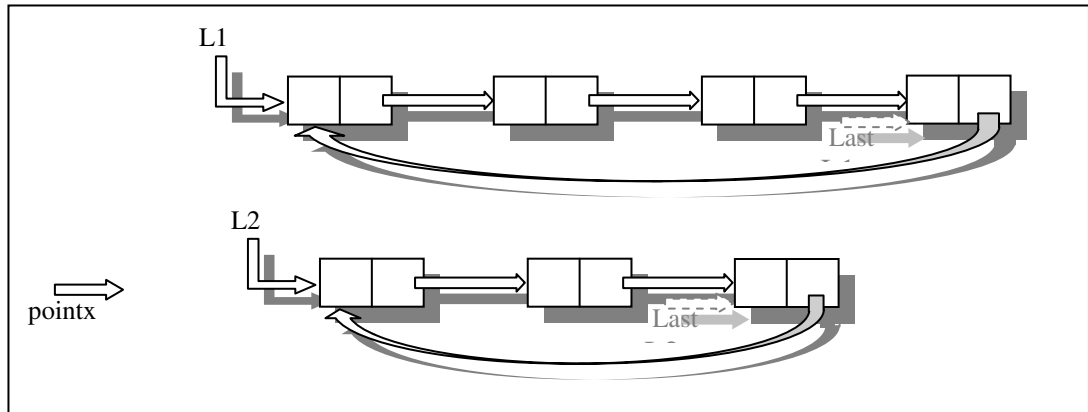
While node[list].link<>list do

Point:= node[point].link;

Last:=point;

End;

Procedure concatenate: Tek bağlı dairesel listelerde iki diziyi birleştirme. Bu procedure'ün kullanımında önemli olan doğru işlem sırasını takip etmektir eğer öncelikli yapılması gereken bağlantılar sonraya bırakılırsa son node'un bulunmasında sorunlar çıkacaktır.



Şekil 4.20. Cuthead ile birleştirilecek L1 ve L2 listeleri

```
Procedure concatenate(var L1:integer;L2:integer);
```

```
Begin
```

```
if L1=nil then L1:=L2
```

```
else if L2=nil then { L2=0 ise L1 e dokunma }
```

```
else
```

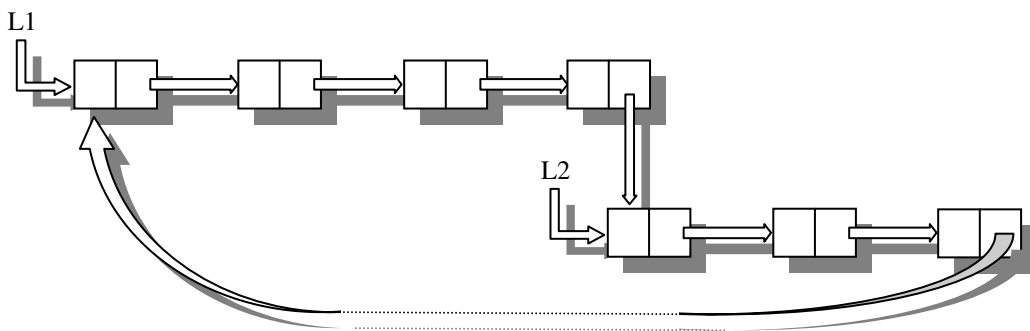
```
begin
```

```
node [last(L1)].Link:=L2;
```

```
node [last(L2)].Link:=L1;
```

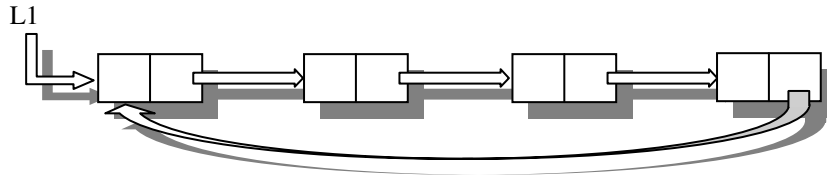
```
end;
```

```
End;
```



Şekil 4.21. Cuthead ile birleştirilmiş liste

Procedure addhead: Tek bağılı dairesel listelerde node ekleme.

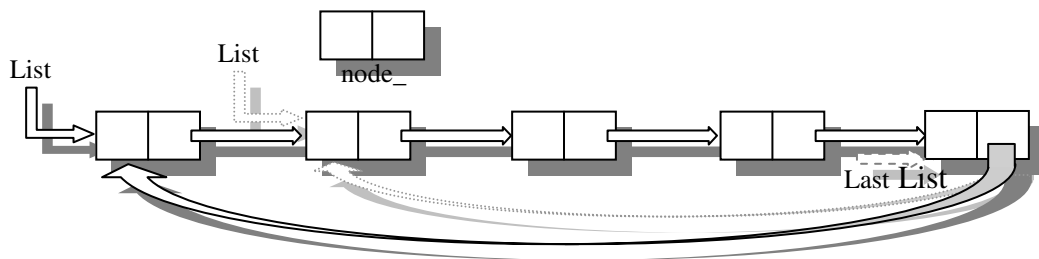


Şekil 4.22. Addhead ile node eklenecek L1 listesi

```

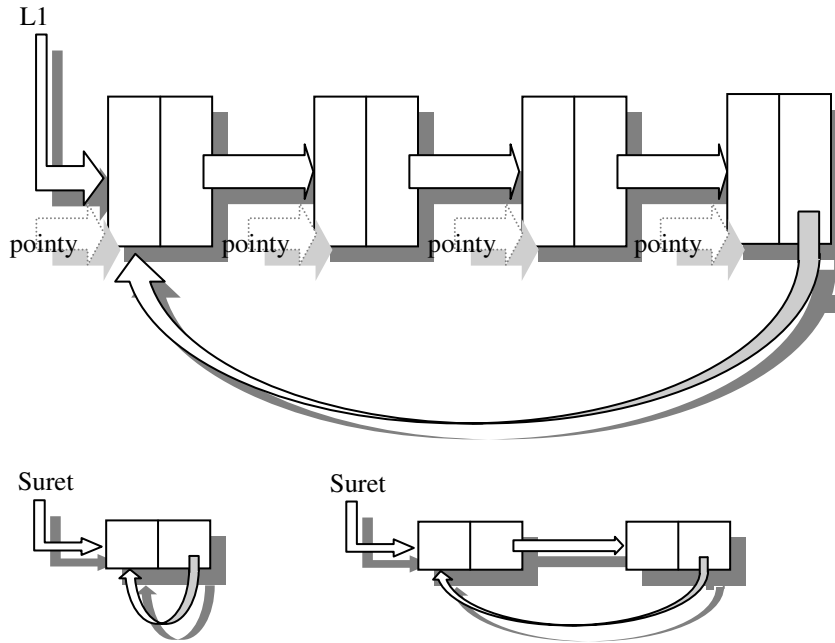
Procedure addhead(node_: integer; var list: integer);
Begin
  If list <> nil Then
  Begin
    Node[node_].link := list;
    Node[last(list)].link := node_;
    List := node_;
  End;
  Else
  Begin
    List := node_;
    Node[node_].link := list;
  End;
End;

```



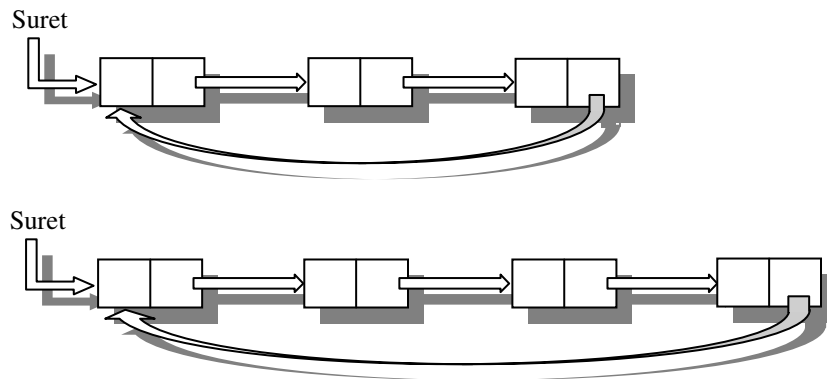
Şekil 4.23. Addhead ile node eklenmiş liste

Function copy: Tek bağılı dairesel listelerde bir dizinin kopyasını elde etme. Copy procedure'ünde cons fonksiyonu yardımıyla avail dizisinden yeni node'lar koparılıp eldeki dizinin node değerleri bu yeni node'lara eşitlenir ve eldeki dizinin kopyası çıkarılmış olur.



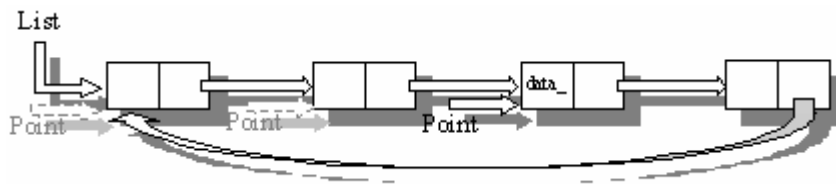
Şekil 4.24. Function copy ile dizi kopyalama

```
Function copy(list: integer): integer;
Var
Suret,point,x: integer;
Begin
Suret:= nill; point:=list;
if list<>nill Then Repeat
x := cons(node[point].data,node[point].link);
Node[x].link:=x;
Concatenate(suret,x);
Point:=node[point].link;
Until point:=List;
Copy:= suret;
End;
```



Şekil 4.25. Function copy ile tamamlanmış kopyalama

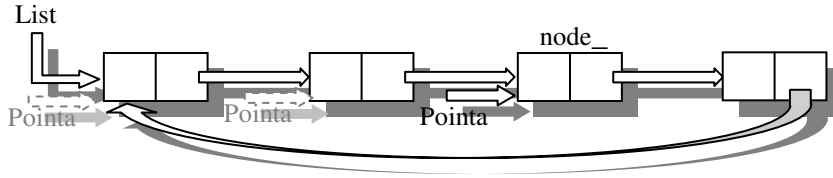
Function locate: Tek bağlı dairesel listelerde bir node'un yerini bulma. Bu function ile bir node'un yeri bulunup data'sına bakılır.



Şekil 4.26. Function locate ile node un yerini bulma

```
Function locate(data_:real, list_: integer): integer;
Var
Point:integer;
Begin
Locate:= nill; { locate ile istediğimizi bulamazsak baştan nill'e eşitliyoruz }
Point:=list;
If point<>nill Then Repeat
If node[point].data<>data_ Then Point:=node[point].link;
Else Begin
Locate:=point;
exit; end;Until point=list; End;
```

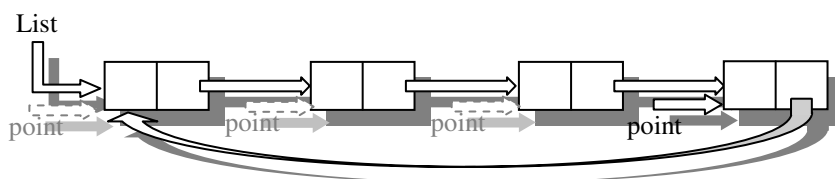
Function member: Tek bağlı dairesel listelerde bir node'u arama



Şekil 4.27. Function member ile node arama

```
Function member(node_, list: integer): boolean;
Var
Point;
Begin
Point:=list;
If point<> nill Then Repeat
If point<>node_ Then Point:= node[point].link;
Until point=list;
Member:= node_=list;
End;
```

Function advance: Tek bağlı dairesel listelerde bir node kadar ilerleme.



Şekil 4.28. Function advance ile bir node ilerleme

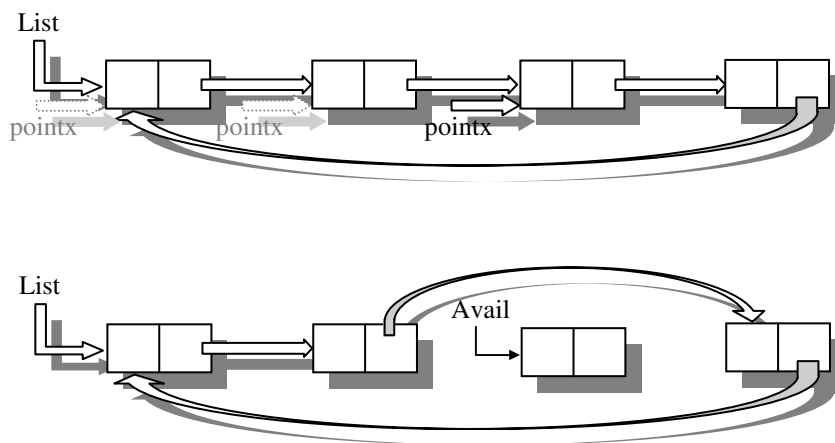
```
Function advance(var point: integer,list:integer): boolean;
Begin
Advance:= false;
If point <> nill and node[point.link]<>list then Begin
If node[point].link <> list Then
```

```

Begin
Point:= node[point].link;
Advance:= true;
End;
End;
End;

```

Function delete: Tek bağlı dairesel listelerde bir node'un silinmesi.



Şekil 4.29. Function delete ile bir node un silinmesi

```

Function delete(node_: integer; var list: integer): boolean;
Var
Point: integer;
Begin
Delete:= false;
If list= nill Then exit;
If list= node_ Then Begin
Addhead(cuthead(list), avail);
Delete:= true;
End;
Else
Begin

```

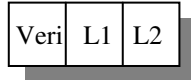
```

Point:= list;
Repeat
If node[point].link= node_ Then Begin
Addhead(cuthead(node[point].link), avail);
Delete:= true;
Exit;
End;
Until not advance(point,list);
End;
End;
Else'den itibaren yazılabilecek alternatif program.
* Else
Begin
Point:= list;
While node[point].link<>list do
Begin
If node[point].link= node_ Then Begin
Addhead(cuthead(node[point].link), avail);
Delete:= true;
Exit;
End;
Else point:=node[point].link
End;
End;

```

4.2. 3. İki bağılı doğrusal listeler

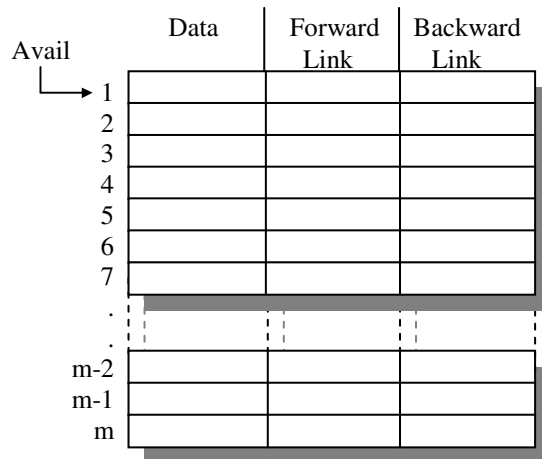
Tek bağılı listelerden farklı olarak forwardlink(ileri yönde bağlantı), backwardlink (geri yönde bağlantı) vardır.



Şekil 4.30. İki bağılı doğrusal listelerde node uzayı

Flink bir sonraki node'u gösterir (forward link).Blink bir önceki node'u gösterir (backward link)

Node uz: Burada aynı tek bağılı listelerde kullandığımız node_uz unitine benzer şekilde bir unit kullanıyoruz. Fakat burada bir tane sonraki node'u gösterecek, bir tane de önceki de'u gösterecek linkleri tanımlıyoruz. Bunların integer tipinde olması yeterlidir. Çünkü node uzayımız yine 10 adet node'dan oluşmaktadır.



Şekil 4.31. İki bağılı doğrusal listelerde boş hafıza alanı

Unit node_uz;

Interface (**)

Cons

Memory=10;

```

Nill=0;
Var
Avail: integer; Node: array[1...memory] of record
Data:real;
Blink:integer;
Flink:integer;
End;
Implementation (**)
End.

```

Procedure dumpmem: Node uzayımızın tüm data ve link'lerini ekrana basarak node uzayında ne olduğunu gösterir.

```

Procedure dumpmem;
Var
i: integer;
Begin
For i:= 1 to memory do
Writeln('node[' ,1:3,']=',node[i].data ' , '
node[i].flink', ' node[i].blink');
End;

```

```

Procedure nodeinit;
Var
i: integer;
Begin
Avail:= 1;
For i:= 1 to memory-1 do
Node[i]. flink:= i+1;
Node[memory]. flink:=nill;
Node[1].blink:=nill;
For i:=memory down to 2 do

```

```
Node[i].blink:=i-1;
```

```
End;
```

	Data	Forward Link	Backward Link
1	Data1	2	Nil
2	Data2	3	1
3	Data3	4	2
4	Data4	5	3
5	Data5	6	4
6	Data3	7	5
7	Data7	8	6
...	...	...	...
m-2	Datam-2	m-1	m-3
m-1	Datam-1	m	m-2
m	Datam	Nil	m-1

Şekil 4.32. İki bağlı doğrusal listelerde dolu hafıza alanı

Function newnode: Newnode fonksiyonunda linklerle ilgili bir işlem yapmadığımızdan dolayı tek bağlı listelerdeki fonksiyondan farklı olarak bir şey yapmıyoruz.

```
Function newnode: integer;
```

```
Var
```

```
New: integer;
```

```
Begin
```

```
New:= cuthead(avail);
```

```
If new= Nil Then Begin
```

```
Writeln('nodeSpace Full...');
```

```
Halt;
```

```
End;
```

```
{Else}
```

```
Newnode:= new;
```

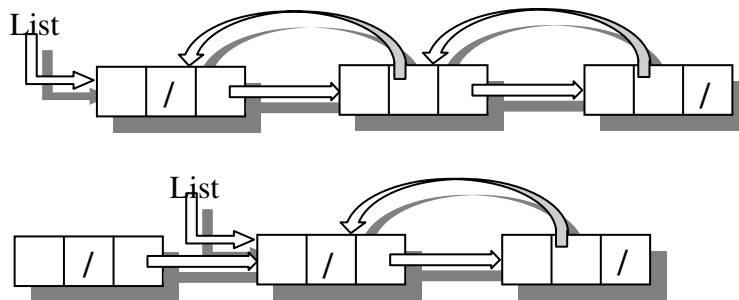
```
End;
```

Function cuthead: Listenin ilk node'unu koparır.

```

Function cuthead(var list: integer): integer;
Begin
Cuthead:= list;
If list <> nill then
If node[list].flink<>nill Then
Begin
Node[node[list].flink].blink:=nill;
List:= node[list].link;
End;
Else list:=nill;
End;

```



Şekil 4.33. Fuction cuthead ile ilk node koparma

Function last: Bu fonksiyon bize listemizdeki en son node'u veriyordu. O halde bu fonksiyonumuzda bir değişiklik yapmamız gerekecek. Link yerlerine flink yazarak bu değişikliği gerçekleştiriyoruz. En sonunda da last :=list diyerek fonksiyonumuzun dönüşte bize list'i göndermesini sağlıyoruz.

```

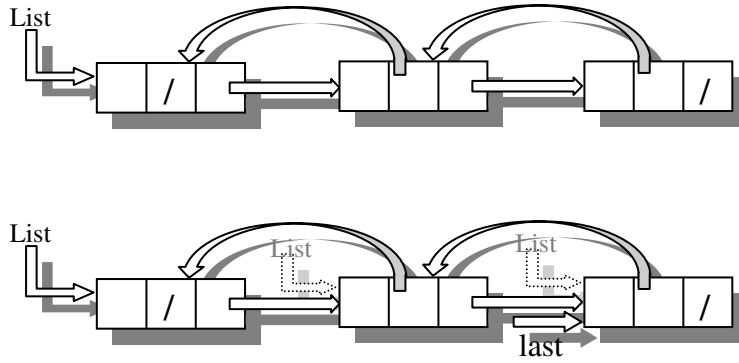
Function last (list: integer): integer;
Begin
If list<> nill Then
While node[list].flink<>nill do

```

```

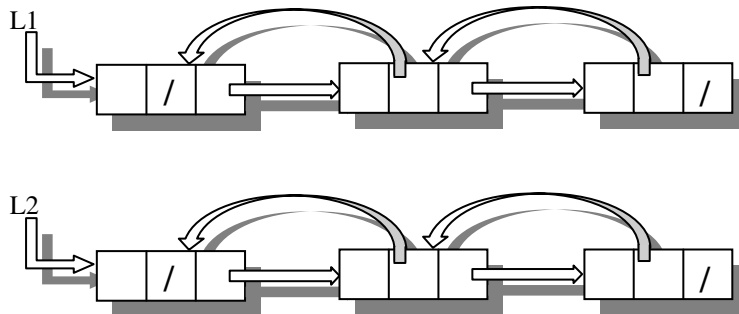
List:= node[list].flink;
Last:=list;
End;

```



Şekil 4.34. Fuction last ile en son node ulaşma

Procedure concatenate: Diyelim ki elimizde iki bağlı L1 ve L2 listeleri var. Bunları birleştirmek için kullandığımız fonksiyondur.



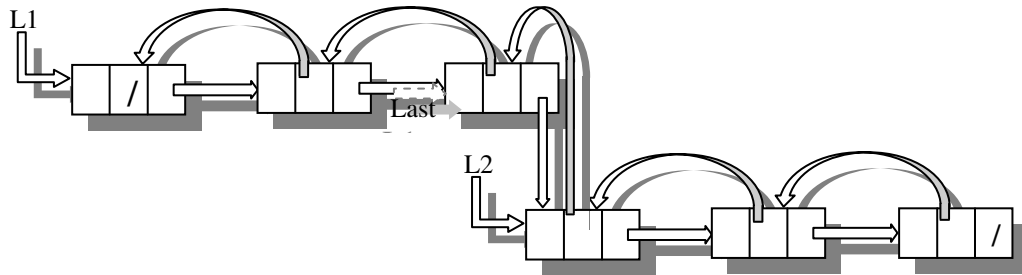
Şekil 4.35. Concatenate ile birleştirilecek L1 ve L2 listeleri

Eğer L1 boşsa L2'yi L1'in arkasına ekliyoruz.. Eğer L1 dolu ise, L1'in en son node'unu buluyoruz. Daha sonra L1'in last'ı L2 olsun diyoruz. Tek bağlı diziden farklı olarak $\text{node}[L2].\text{blink} := \text{list}(L1)$ dir.

```

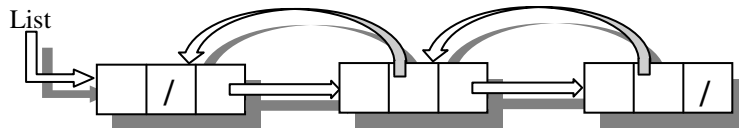
Procedure concatenate(var L1: integer; L2: integer);
Begin
  If L2<>nill Then
    If L1=nill Then  L1:=L2;
    Else Begin
      Node[L2]. Blink:=last[L1];
      Node[last(L1)]. Flink:=L2;
    End;
  End;
End;

```



Şekil 4.36. Concatenate ile birleştirilmiş liste

Procedure free: Bu procedure kullanmadığımız diziyi avail dizisine eklemek için kullanıyoruz. Bu fonksiyon tek bağlı dizi yapısındaki free procedure'ü ile aynı yapıdadır.

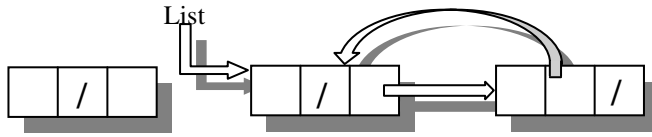


Şekil 4.37. Procedure free ile diziyi avail dizisine ekleme

```

Procedure free(list: integer);
Begin
  Concatenate(list,avail);
  Avail:= list;
End;

```



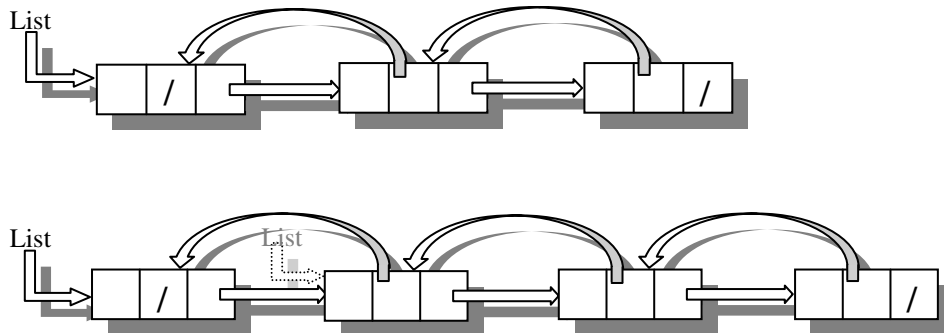
Şekil 4.38. Procedure free ile avail eklenmiş dizi

Procedure addhead: İki bağlı doğrusal listelerde diziye istenen node'un eklenmesini sağlayan program.

```

Procedure addhead(node_: integer; var list: integer);
Begin
  If list:=nill Then Begin
    Node[node_].fblink:= nill;
    Node[node_].blink:=nill;
  End;
  Else Begin
    Node[list].blink:=node_;
    Node[node_].fblink:=list;
    Node[node_].blink:=nill;
  End;
  List:=node_;
End;

```



Şekil 4.39. Procedure addhead ile diziye node ekleme

Procedure cons: Newnode ile yeni bir node alınıp cons'a gönderilen data ve link değerleri bu node'a aktarılıyor.

```
Function cons (data_:real; flink_,blink_:integer):integer;
Var
Cons_: integer;
Begin
Cons_:newnode;
Cons: cons_;
Node[cons_].data:=data_;
Node[cons_].flink:=flink_;
Node[cons_].blink:=blink_;
End;
```

Function copy: İki bağlı doğrusal listelerde bir dizinin kopyasını elde etmeyi sağlayan program.

```
Function copy(list: integer): integer;
Var
Suret,point: integer;
Begin
Suret:= nill;
Point:=list;
If list<>nill Then
Repeat
Concatenate(suret, cons(node[point].data, nill,nill));
point:= node[point].flink;
Until point= nill;
Copy:= suret;
End;
End.
```

Function locate: İki bağlı doğrusal listelerde aranan node'un yerini bildiren program.

Data_: Data arıyoruz ve bulunursa data_'nın olduğu nodun numarasını geri döndürüyor.

```
Function locate(data_, list_: integer): integer;

Begin
  Locate:= nil;
  While list <> nil do
    If node[list].data <> data_ Then list:=node[list].flink
  Else Begin
    Locate:= list;
  Exit;
  End;
  End;
```

Function member: İki bağlı doğrusal listelerde istenilen node'un var olup olmadığını bildiren program.

```
Function member(node_, list: integer): boolean;

Begin
  While (list <> nil) and (list <> node_) do
    List:= node[list].flink;
  Member:= list=node_;
  End;
```

Function advance: İki bağlı doğrusal listelerde bir node ileri gitmeyi sağlayan program.

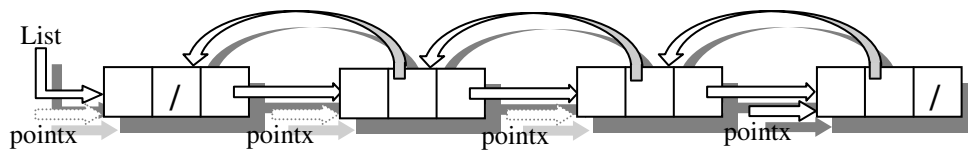
```
Function advance(var point: integer): boolean;

Begin
  Advance:= false;
```

```

If point <> nil Then
If node[point].flink <> nil Then
Begin
Point:= node[point].flink;
Advance:= true;
End;
End;

```



Şekil 4.40. Function advance ile bir node ilerleme

Function delete: İki bağlı doğrusal listelerde istenilen node'u silen program.

```

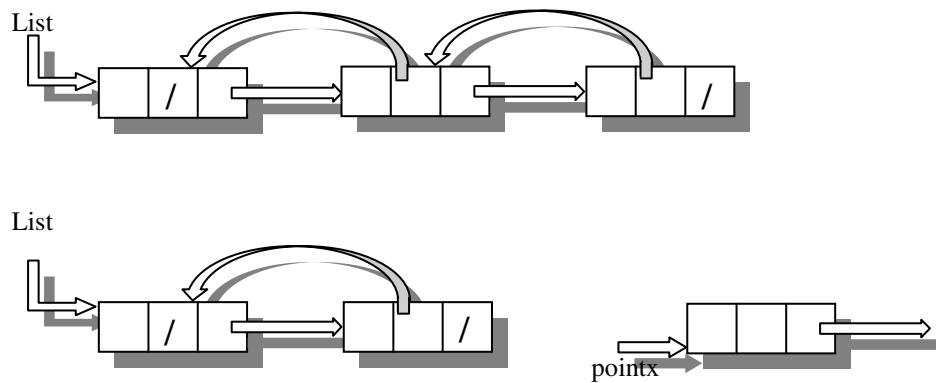
Function delete(node_: integer; var list: integer): boolean;
Var
Point: integer;
Begin
Delete:= false;
If list= nil Then exit;
If list= node_ Then Begin
Addhead (cuthead(list), avail);
Delete:= true;
End;
Else Begin
Point:= list;
While node[point] .flink <> nil do
Begin
if node[point].flink= node_ then
Begin
if node[node[point].flink].flink=nil then begin

```

```

node[point].flink:=null;
addhead(node[point].flink,avail);
delete:=true;
exit;
end;
else begin
point2:=node[point].flink;
node[point].flink:= node[node[point].flink].flink;
node[node[point].flink].blink:=point;
addhead(point2,avail);
delete:=true;
exit;
end;
Else point:= node[point] .flink;
End;
End;
End ;

```



Şekil 4.41. Function delete ile istenilen node silme

4. 2. 4. İki bağlı dairesel listeler

Unit node_uz

```
Unit node_uz;
Interface (**)
Cons
Memory=10;
Nill=0;
Var
Avail: integer;
Node: array[1...memory] of record
Data:real;
Blink:integer;
Flink:integer;
End;
Imple. mentation (**)
End
```

Procedure dumpmem:

```
Procedure    dumpmem;
Var
i: integer;
Begin
For i:= 1 to memory do
Writeln('node[' ,1:3,']=' ,node[i].data ' , ' node[i].flink', ' node[i].blink');
End;
```

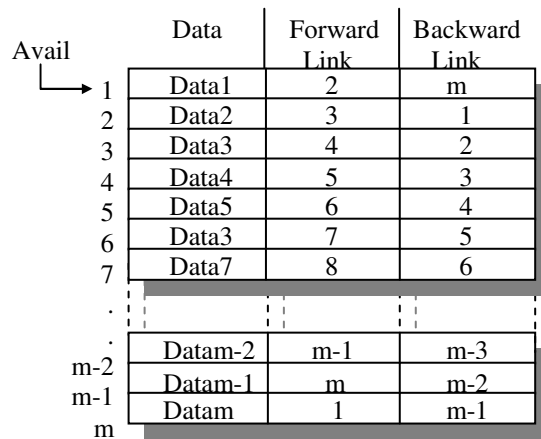
Procedure nodeinit:

İki bağlı dairesel listelerde node'ları birbirlerine bağlama.

```

Procedure nodeinit;
Var
i: integer;
Begin
Avail:= 1;
For i:= 1 to memory-1 do
Node[i]. flink:= i+1;
Node[memory]. flink:=1;
Node[1].blink:=memory;
For i:=memory down to 2 do
Node[i].blink:=i-1;
End;

```



Şekil 4.42. İki bağlı dairesel listelerde dolu hafıza alanı

Function newnode: İki bağlı dairesel listelerde listeye yeni bir node eklemeye yarayan program.

```

Function newnode: integer;
Var
New: integer;
Begin
New:= cuthead(avail);

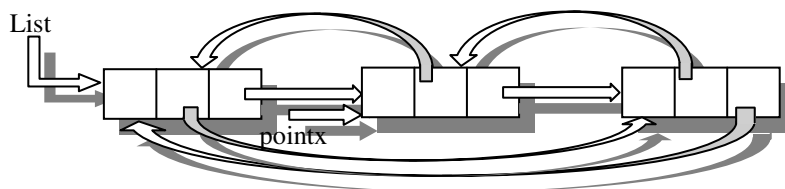
```

```

If new= Nill Then
Begin
Writeln('nodeSpace Full...');
Halt;
End; {Else}
Newnode:= new;
End;

```

Function cuthead: İki bağlı dairesel listelerde dizinin ilk node'unu silen program.

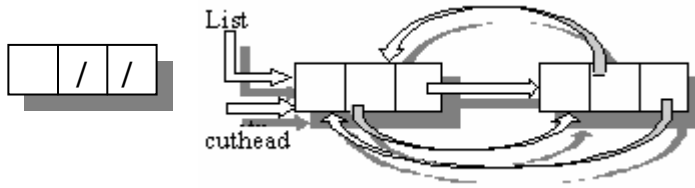


Şekil 4.43. İki bağlı dairesel listelerde dizinin ilk node unu silme

```

Function cuthead(var list: integer): integer;
Begin
Cuthead:= list;
If list <> nill Then
If node[list]. flink=list then liet:=nill
Else
Begin
Node[node[list]flink].blink:= node[list]. blink;
Node[node[list]blink].flink:= node[lisy]. flink;
List:=node[list]. flink;
End;
End;

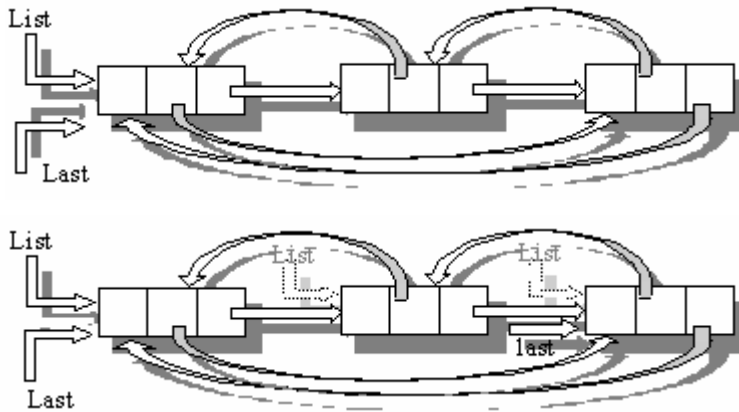
```



Şekil 4.44. İki bağlı dairesel listelerde ilk node silinmiş dizi

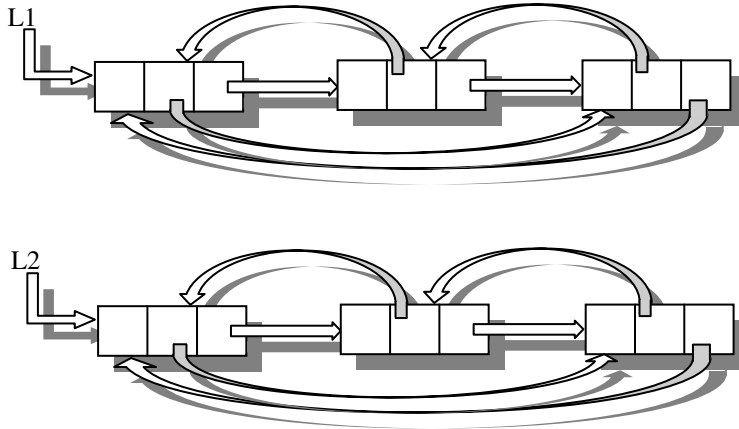
Function last: İki bağlı dairesel listelerde dizinin son node'unu bulan program.

```
Function last (list: integer): integer;
Begin
Last:=list;
If list<> nill Then
Last:=node[list]. blink;
End;
```



Şekil 4.45. İki bağlı dairesel listelerde son node bulma

Function concatenate: İki bağlı dairesel listelerde iki diziyi birleştiren program.

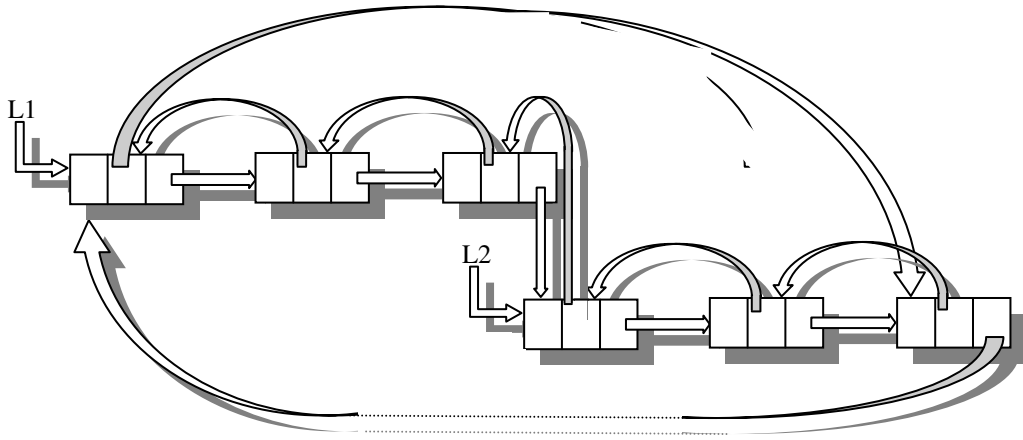


Şekil 4.46. İki bağlı dairesel listelerde L1 ve L2 listelerini birleştirme

```

Procedure concatenate (var L1:integer; L2:integer);
Var
Point:integer;
Begin
If (L2<>nill) Then if (L1=nill) then L1:=L2 Else *   Begin
Point:=last(L1);
Node[last(L1)].flink:=L2;
Node[last(L2)].flink:=L1;
Node[L1].blink:=last(L2);
Node[L2].blink:=point;
End;
End;
*'dan itibaren alternatif program.
Begin
Point:= Node[L1].blink;
Node[node[L1].blink].flink:=L2;
Node[node[L1].blink].flink:=L1;
Node[L1].blink:= node[L2].blink;
Node[L2].blink:=point;
End;
End;

```



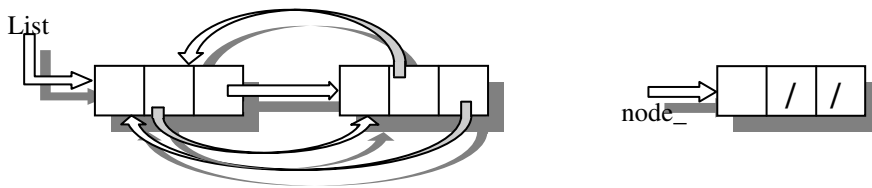
Şekil 4.47. İki bağlı dairesel listelerde birleştirilmiş liste

```

Procedure free(list: integer);
Begin
Concatenate(list,avail);
Avail:= list;
End;

```

Procedure addhead: İki bağlı dairesel listelerde diziye yeni bir node ekleyen program.



Şekil 4.48. İki bağlı dairesel listelerde diziye yeni node ekleme

```

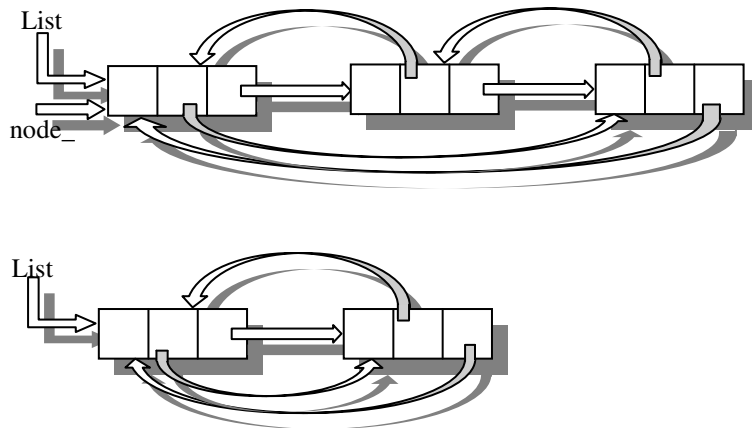
Procedure addhead(node_: integer; var list: integer);
Begin
If list=nil Then Begin
Node[node_].flink:= nil;
Node[node_].blink:=nil;
End;
End;

```

```

Else
Begin
Node[node[list].blink].flink:=node_;
Node[node_].flink:=list;
Node[node_].blink:=node[list].blink;
Node[list].blink:=node_;
End;
List:=node_;
End;

```



Şekil 4.49. İki bağlı dairesel listelerde yeni node eklenmiş dizi

Function cons: İki bağlı dairesel listelerde dizinin içeriğini tanımlayan program.



Şekil 4.50. İki bağlı dairesel listelerde dizi tanımı

```

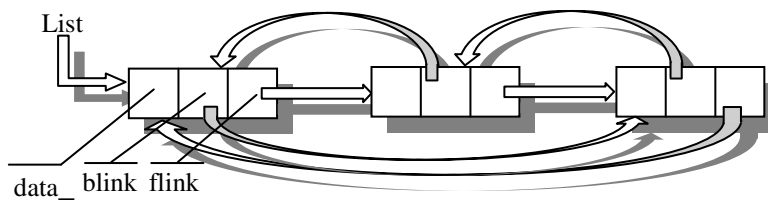
Function cons (data_:real; flink_.blink_:integer):integer;
Var
Cons_: integer;
Begin
Cons_:newnode;

```

```

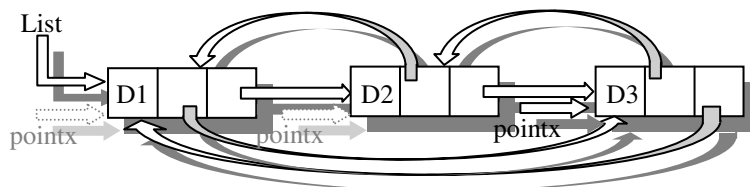
Cons: cons_;
Node[cons_].data:=data_;
Node[cons_].flink:=flink_;
Node[cons_].blink:=blink_;
End;

```



Şekil 4.51. İki bağlı dairesel cons işlemi

Function copy: İki bağlı dairesel listelerde elimizdeki listenin kopyasını çıkarmamızı sağlayacak program.



Şekil 4.52. İki bağlı dairesel kopyalanacak liste

```

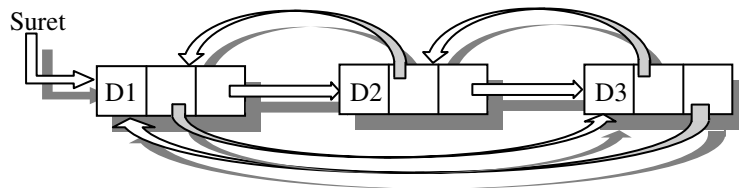
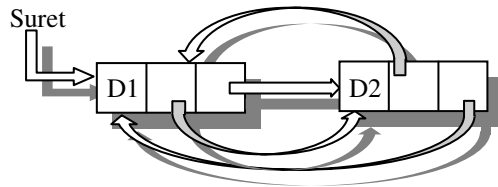
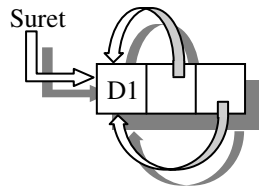
Function copy(list: integer): integer;
Var
Suret,point: integer;
Begin
Suret:= nill;
Point:=list;
If list<>nill Then
Repeat
Concatenate(suret,cons(node [list].data, nill, nill));
Point:=node[point].flink;

```

```

Until point:=list;
Copy :=suret;
End;

```



Şekil 4.53. İki bağlı dairesel listelerde kopyalama işlemi

Function locate: İki bağlı dairesel listelerde bir data'nın yerini bulmamızı sağlayan program.

```

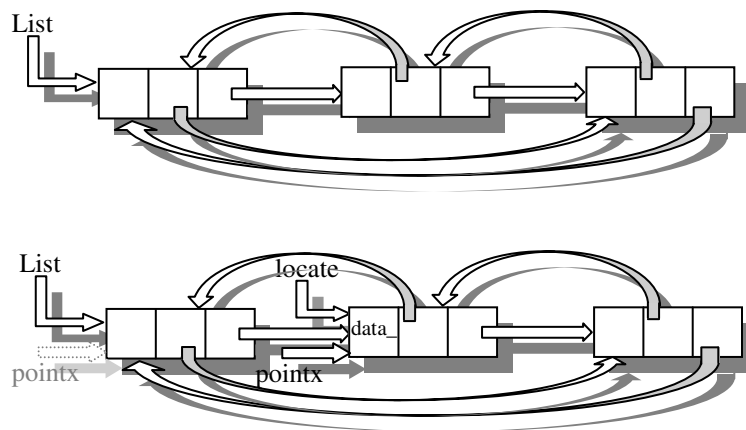
Function locate (data_: real; list: integer): integer;
Var
Point:integer;
Begin
Locate:=nill; point:=list;
If list<>nill Then
Repeat
If node[point].data<>data_ Then
point:=node[point].flink;
Else

```

```

Begin
Locate:=point;
Exit;
End;
Until point:=list;
End;

```



Şekil 4.54. İki bağlı dairesel listelerde data nın yerini bulma işlemi

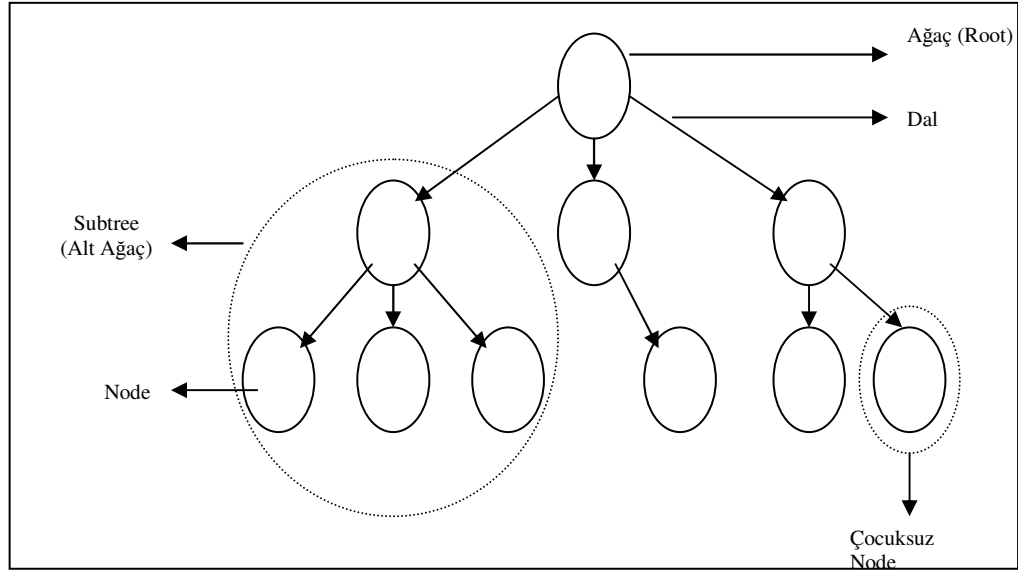
Function member:İki bağlı dairesel listelerde bir node'nin yerini bulmamızı sağlar.

```

Function member(node_, list: integer): boolean;
Var
Point:integer;
Begin
Point:=list;
If list<>nill Then
Repeat
If point<>node_ Then point:=node[point].fnext;
Until (point=list) or (point=node_);
Member:= point=node_;
End;

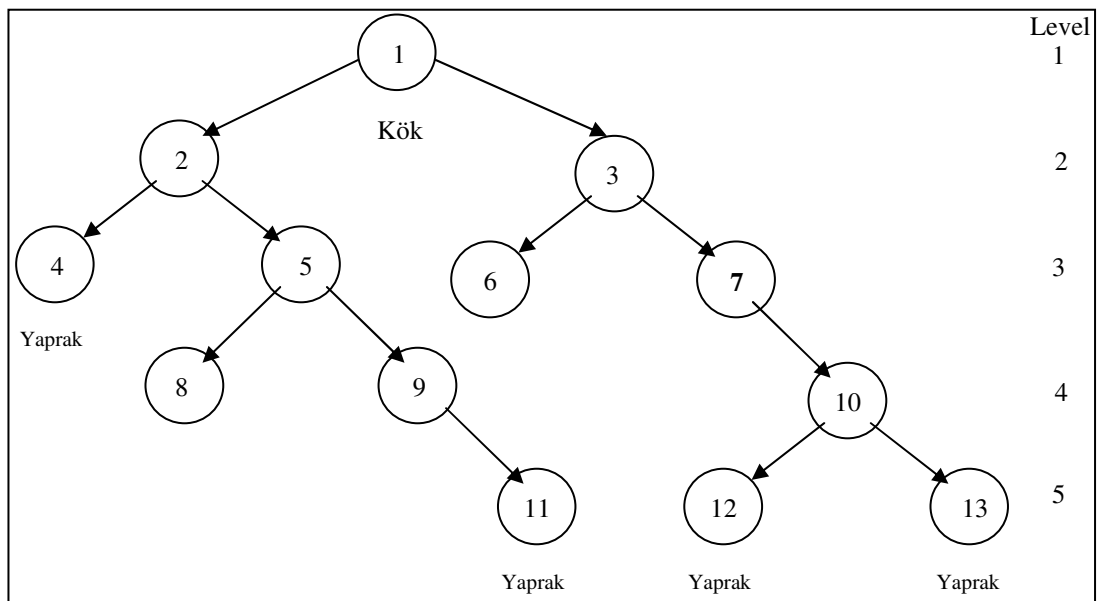
```

4.3. Ağaçlar

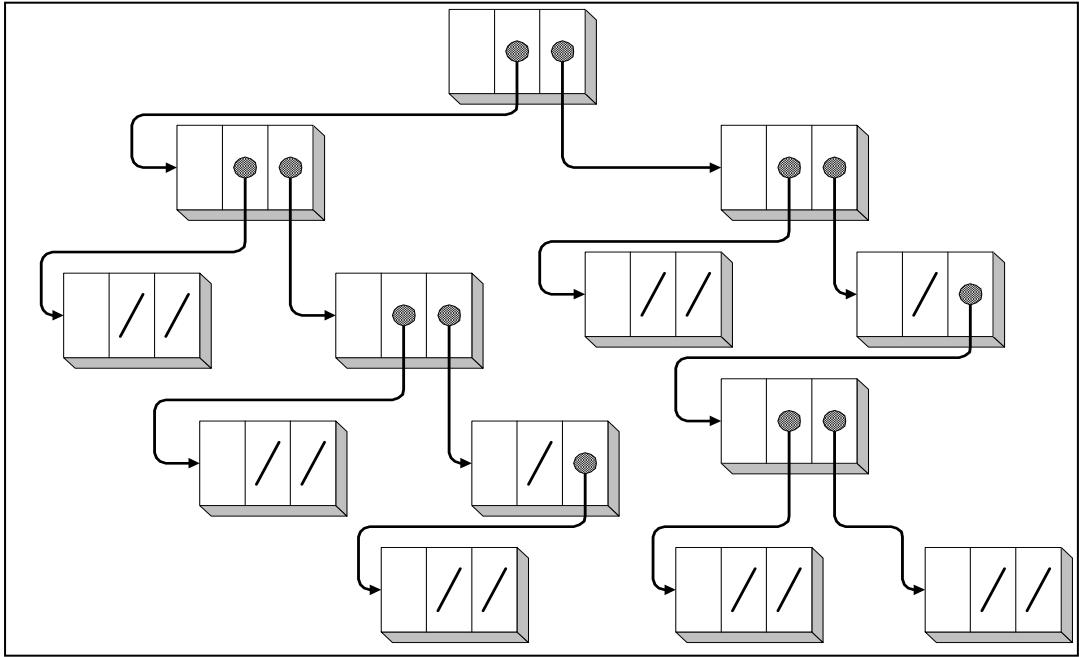


Şekil 4.55.(a) Ağaç Yapısı

Bir Binary ağaçta, bir seviyedeki maximum node sayısı 2^{l-1} 'dir. ($l \geq 1$ olmalıdır). K derinliğine sahip bir binary ağaçta maximum node sayısı $2^K - 1$ 'dir. ($K \geq 1$ olmalıdır). Seviye ve derinlik kökle başlayacak şekilde aynı sayılmaktadır. Ağaçta nodeların eksik kısımları null olarak adlandırılır.

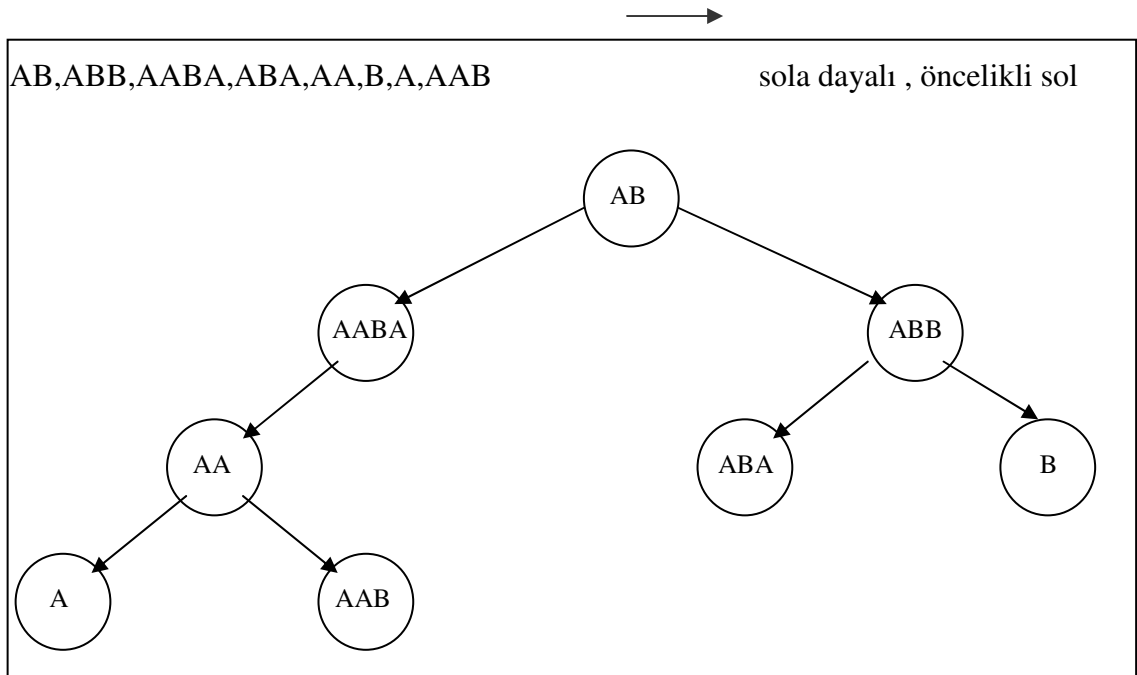


Şekil 4.55.(b) Ağaç Yapısı



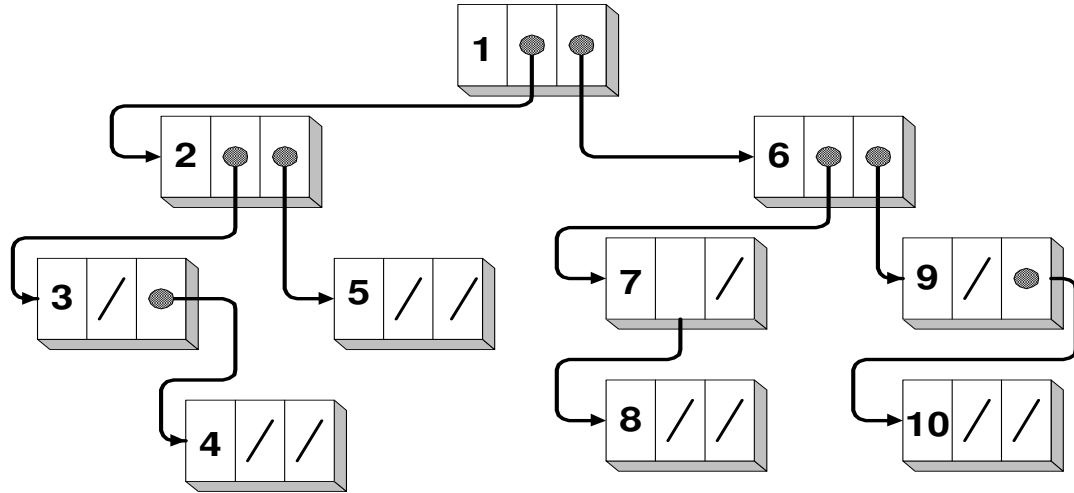
Şekil 4.55. (c) Ağaç Yapısı

Soru: Aşağıdaki verilen geliş sıralarına göre nasıl bir sıralama ağacı oluşturacaklarını iki bağlı node'ların kullanıldığı bir ikilik ağaç çizerek gösteriniz.



Şekil 4.56. İki bağlı node ların kullanıldığı ikilik ağaç

Printree Procedure'ü:



Şekil 4.57. Printree procedure ü yapısı

```

Procedure Printree (t:integer);
Begin
  If T<>nill then
  Begin
    If node[T].link<>nill Printree(node[T].llink);
    Writeln (node[T].data);
    If node [i].Rlink<>nill Printtree(node[T].Rlink);
  End;
End;

```

Bir ağaçtaki sıralama şekilleri:

Inorder: Left tree'nin en sol ucundan başlayarak sağa tarama yapılır.

Preorder :Önce ana node sonra yan node'lar okunur.

Postorder: Sondan başa doğru sol öncelikli olarak tarama yapılır.

In Order:

```

Procedure Printtree(t:integer);

```

```

Begin
If t<>nill Then
Begin
If node[t].link <> printtree(node[t].link);
Writeln (node[t].data);
If node[t].rlink <> nill printtree(node[t].rlink);
End;
End;

```

Pre order:

```

Procedure Printtree(t:integer);
Begin
If t<>nill Then
Begin
Writeln (node[t].data);
If node[t].link <> printtree(node[t].link);
If node[t].rlink <> nill printtree(node[t].rlink);
End;
End;

```

Post order :

```

Procedure Printtree(t:integer);
Begin
If t<>nill Then
Begin
If node[t].link <> printtree(node[t].link);
If node[t].rlink <> nill printtree(node[t].rlink);
Writeln (node[t].data);
End;
End;

```

5. SONUÇ VE ÖNERİLER

Bilgi Çağı eğitim kurumları öğrencilerine dersliklerde belli zaman dilimlerinde sınırlı bilgi aktarmak gibi klasik ya da gelenekselleşmiş yapıdan uzaklaşmak zorundadır. Bilgi Çağı öğrencileri kendilerinden çok şey beklenen ve istenilen öğrenciler olacağından çok iyi öğrenmek, sürekli öğrenmek ve öğrendiklerini güncellemek zorundadırlar. Bu yüzden eğitim ve öğretimde bilişim teknolojilerinin ve uzaktan eğitim sistemlerinin yaygınlaştırması gerekmektedir [9].

Bilgisayar destekli eğitim günden güne önem kazanmaktadır. Bu nedenle bilgisayar destekli eğitim materyalleri de sürekli değişmektedir. Bir zamanlar sadece metinden oluşan materyaller, günümüzde iletişim, ses ve görüntü transferi, video konferans gibi Internet aracılığı ile kullanılabilen servisler ile zenginleşmektedir. Ayrıca böyle bir örneğin sunulması eğitim materyalinin içeriğini zenginleştirecek ve görsel olarak öğrenebilme stiline sahip kullanıcılar için büyük bir avantaj sağlayacaktır [10].

Internet tabanlı eğitim yalnızca bir web programcısının hazırlayacağı sayfalardan ibaret değildir. Aksine uzun ve zahmetli bir hazırlık dönemi gerektiren, işin gerekleri tam yerine getirildiğinde bir anlam kazanan eğitim şeklidir [10].

Son yüzyılda eğitim ve öğretimi Internet olmadan düşünmemiz çok yanlış olur. Web tabanlı eğitim için Internet'i, bilgi aktarma, arama ve geliştirme aracı olarak kullandık. Hazırladığımız web sitesi ile Veri yapıları Dersinin daha iyi anlaşılmasını ve verimin yükselmesini sağlamaya çalıştık.

Hazırladığımız, web sitesinin tasarımı ve Veri Yapıları Dersinin içeriği konusunda bazı eğitim ve öğretimcilere ait görüş ve değerlendirmeler:

Öğr.Gör.Ayhan Tetik(Ondokuz Mayıs Üniversitesi):

“Veri yapıları dersinin internet ortamından aktarılması uygulaması teknik eğitim sahasındaki kaynakların yeterince olmadığı bir alanda büyük bir boşluğu doldurmaktadır. Özellikle teknik eğitimde zorunlu olan derse devamı ortadan

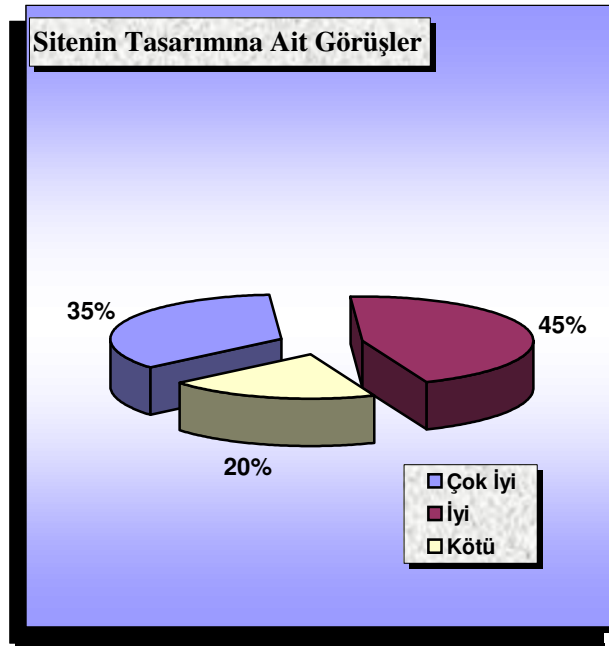
kaldırmakta ve bu konuda eğitim alanlara rahatlık sağlamaktadır. Web sitesi, animasyonlarla biraz daha göze hitap eder hale getirilip, ders canlı olarak aktarılıp, bağlantı sağlanabilirse daha da faydalı olacağı kanaatindeyim. Bu çalışmada emeği geçenleri tebrik ediyor, çalışmalarını daha da güzele götürmelerini diliyorum.”

Öğr.Gör.Recep Kılınç(Ondokuz Mayıs Üniversitesi):

“Program geliştirme ve program tasarımında en önemli konu, ele alınan uygulamaya en uygun veri modelinin belirlenmesi, veri yapısının tanımlanması olayıdır. Veri yapısı, verinin bellekte tutulma şeklini gösterir. Veri modeliyse, verilerin birbirleriyle ilişkisini ve sırasal pozisyonunu gösterir. Hazırladığınız bu eğitim amaçlı web sitesinde veri yapıları ile ilgili gerekli bilgilerin çoğuna değinilmiştir. Bu yüzden bu konuda bu sitenin önemli bir boşluğu doldurduğu düşüncesindeyim.”

Web sitesinin tasarımı konusunda Meslek Yüksekokulu Bilgisayar Programcılığı bölümü öğrencilerine yapılan anket değerlendirme sonucu:

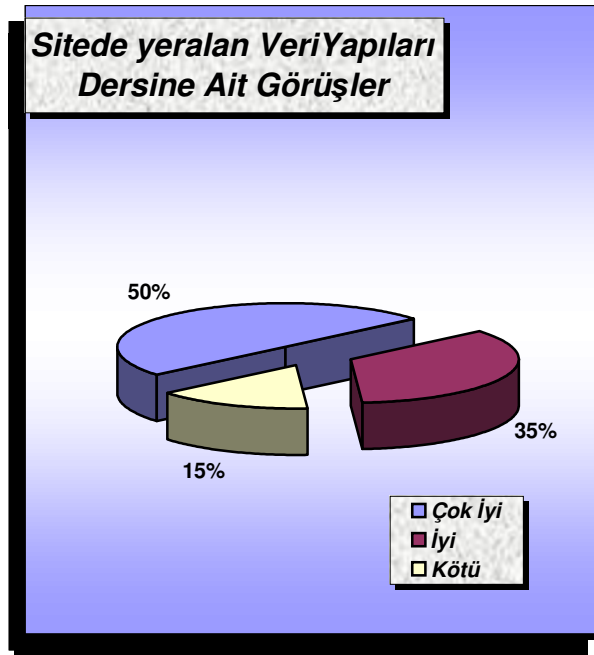
Ankete katılan öğrenci sayısı: 20



Resim 5.1. Web sitesinin tasarımına ait anket sonucu

Web sitesinde yayınlanan Veri Yapılarının içeriği konusunda Meslek Yüksekokulu Bilgisayar Programcılığı bölümü öğrencilerine yapılan anket değerlendirmesi sonucu:

Ankete katılan öğrenci sayısı: 20



Resim 5.2. Veri Yapıları Dersi nin içeriğine ait anket sonucu

Sonuç olarak Internet üzerinden eğitimin sağladığı büyük imkanlar ve kolaylıklar sayesinde eğitime büyük ölçüde yararlar sağlanmıştır. Web tabanlı olarak yapılan eğitim programlarıyla bir çok kişinin eğitim alması, günümüz iletişim ve bilişim araçlarıyla artık kolayca yapılabilir. Bunun küçük bir örneğini bu çalışmada gerçekleştirmeye çalıştık.

KAYNAKLAR

1. Internet: Teknotürk, Online eğitim., <http://www.teknoturk.org/docking/yaziler/tt000042-yazi.htm>, (2001).
2. Gürkan O., FrontPage 2002 ,1. baskı, editörü, Rahime Demir,**Hayat Yayıncılık**, İstanbul, 3-9, (2002).
3. Prof. Dr. Uysal M., PHP 4 ile Web Tasarımı, 1.basım, **Beta Basım Yayın Dağıtım** İstanbul, 1-390, (2001).
4. Baysal G., Photoshop 5.0, 1.baskı, **Pusula yayıncılık ve iletişim** , İstanbul, 23-33, (1992).
5. Özkan Y., Dondurmacı G. A., JavaScript, 1.basım, **Alfa Yayınları**, İstanbul, 10-150, (2001).
6. Gürkan O., Macromedia Fireworks 8 , 1.basım, **Nirvana Yayınları**, İstanbul, 2-50, (2006).
7. Balaban E.,Web Tasarım Klavuzu, 1. basım, **Pusula Yayıncılık**, İstanbul, 20-400, (2003).
8. Şamlı M., PHP ile Web Programcılığı, 2.baskı, **Acar Matbaası Pusula Yayıncılık ve iletişim**, İstanbul, 25-150, (2006).
9. Varol , A.; Varol, N., Bilgi Teknolojilerine Dayalı Uzaktan Yükseköğretim ve Ders Hazırlama İlkeleri Üzerine Öneriler, **BTİE'99, Bilişim Teknolojileri Işığında Eğitim Konferansı ve Sergisi**, Ankara, Bildiriler Kitabı, 15-17, (1999).
10. Türel, Y. (2002), Çevrimiçi Uzaktan Eğitimde İletişim Modülü, **Fırat Üniversitesi, Fen Bilimleri Enstitüsü, Seminer Çalışması**, Elazığ, (2002).
11. Dr.Çölkesen R., Veri Yapıları ve Algoritmalar, 1.baskı, **Papatya Yayıncılık Eğitim A.Ş.** ,İstanbul,155-170, (2004).
12. Thomas A.Powell, HTMLve XHTML, 1.nd ed, **Alfa Yayınları**, İstanbul,10-80, (2004).
13. Dr. Kurnaz S., Veri Yapıları ve Algoritma Temelleri, 1. baskı, **Papatya Yayıncılık Eğitim** , İstanbul,50-200, (2004).
14. Menascé D.A, Almeida V.A.F, Capacity Planning for Web Performance, **Prentice, Hall Inc.**, 15-85, (1998).

EKLER

EK-1 Web sitesinin yapılmasında kullanılan bazı HTML kodları

Index.htm

```
<html>
<head>
<meta name="GENERATOR" content="Microsoft FrontPage 5.0">
<meta name="ProgId" content="FrontPage.Editor.Document">
<meta http-equiv="Content-Type" content="text/html; charset=windows-1254">
<title>...:veri yapıları dersi:...</title>
<style>
<!--
a
{ text-decoration:none;font-family:arial;font-size:8pt;}
-->
</style>
</head>
<body bgcolor="#04A6C8" text="#FFFFFF">
<div align="center">
  <center>
    <table border="15" cellpadding="0" cellspacing="0" style="border-collapse:
collapse" bordercolor="#CC3300" width="313" height="413" id="AutoNumber1"
bgcolor="#006697">
      <tr>
        <td width="702" height="128" colspan="3">
          </td>
        </tr>
        <tr>
          <td width="702" height="33" colspan="3" bgcolor="#F4007A" nowrap
bordercolor="#000000">
            <p align="right"><b><font color="#FFFFFF">
              <script language="JavaScript">
```

EK-1 (devam) Web sitesinin yapılmasında kullanılan bazı HTML kodları

```
<!--
// ismail ipekgoz
function makeArray()
{ var args = makeArray.arguments;
  for (var i = 0; i < args.length; i++)
  { this[i] = args[i]; }
  this.length = args.length; }
function fixDate(date)
{ var base = new Date(0);
  var skew = base.getTime();
  if (skew > 0)
  date.setTime(date.getTime() - skew); }
function getString(date)
{ var months = new makeArray("Ocak", "Subat", "Mart",
  "Nisan", "Mayis", "Haziran",
  "Temmuz", "Agustos", "Eylül",
  "Ekim", "Kasim", "Aralik");
  var days = new makeArray("Pazar", "Pazartesi", "Sali", "Çarsamba",
  "Persembe", "Cuma", "Cumartesi");
  return date.getDate() + " " +
  months[date.getMonth()] + " " +
  (date.getYear()+ " " +
  (days[date.getDay()])); }
var cur = new Date();
fixDate(cur);
var str = getString(cur);
document.write(str);
// -->
</script><span style="position: absolute; left: 145; top: 165">
```

EK-1 (devam) Web sitesinin yapılmasında kullanılan bazı HTML kodları

```
<marquee style="font-size: 10pt; color: #FFFF00; font-weight: bold; font-family: Verdana" width="380" height="15" bgcolor="#4040FF">VERİ YAPILARI DERSİNİN İNTERNET ÜZERİNDEN
```

```
VERİLMESİ</marquee></span></font></b></td>
```

```
</tr>
```

```
<tr>
```

```
<td width="133" height="42" bgcolor="#0083C1">
```

```
<p align="center"><b>
```

```
<font size="2" face="Times New Roman" color="#FFFFFF">&nbsp;<a target="_self" href="index.htm"></a></font></b></td>
```

```
<td width="493" height="247" rowspan="12" bgcolor="#0083C1">
```

```
<iframe name="I1" width="489" height="735" src="main.htm" border="1" frameborder="1" style="border:5px outset #CC3300; ">
```

Tarayıcınız satır içi çerçeveleri desteklemiyor veya şu anda satır içi çerçeveleri göstermek için yapılandırılmamış.</iframe>

```
<p align="center"><b><font size="2" color="#FFFFFF">Copyright © 2006&nbsp;</font>
```

```
<font color="#CC3300" size="2">terfi ve terlif hakkı saklıdır</font><font size="2" color="#FFFFFF">
```

```
&quot;Bu site 1024x768 pikselde çalışmaktadır&quot;</font></b></p>
```

```
<p>&nbsp;</td>
```

```
<td width="100" height="100%" rowspan="6" bgcolor="#FFB3B3">
```

```
<a target="I1" href="video/a.htm">
```

```
</a><p align="left">
```

```
<marquee direction=up width=150 height=100 scrollamount=2  
scrolldelay=0 onMouseOver='this.stop()' onMouseOut='this.start()' style="font-family: Verdana; font-size: 8pt; font-weight: bold; color:#FFFF00"
```

```
bgcolor="#373737">Veri Yapıları Dersinin internet ortamına taşınmasındaki amaç e-
```

EK-1 (devam) Web sitesinin yapılmasında kullanılan bazı HTML kodları

egitim platformunda öğrencilere interaktif bir eğitim vermektir.Böylece
öğrencilerin istedikleri anda ders içeriğine ulaşmaları sağlanmıştır.

```

</tr>
<tr>
  <td width="133" height="42" bgcolor="#0083C1">
    <p align="center">&nbsp;<a href="derstanitimi.htm" target="I1"></a></td>
</tr>
<tr>
  <td width="133" height="8" bgcolor="#0083C1">
    <p align="center"><b>
<font size="2" face="Times New Roman" color="#FFFFFF">&nbsp;<a
target="I1" href="dersamac.htm"></a></font></b></td>
</tr>
<tr>
  <td width="133" height="42" bgcolor="#0083C1">
    <p align="center">
<a href="derskonu.htm" target="I1">
</a></td>
</tr>
<tr>
  <td width="133" height="42" bgcolor="#0083C1">
    <p align="center">
<a target="I1" title="Bu bölümde Veri yapılarıyla ilgili çözümlü örneklerle
ulaşabilirsiniz..." href="diornekler.htm">
</a></td>
</tr>
<tr>

```

EK-1 (devam) Web sitesinin yapılmasında kullanılan bazı HTML kodları

```
<td width="133" height="42" bgcolor="#0083C1">
    <p align="center">
        <a target="I1" title="Bu bölümde Anlamadığınız konuları arkakaşlarınızla
tartışarak o kunuyu pekiştirebilirsiniz..." href="degerlen.htm">
            </a></td>
</tr>
<tr>
    <td width="133" height="42" bgcolor="#0083C1">
        <p align="center">
            <a href="anket.htm" target="I1">
                </a></td>
            <td width="149" height="42" bgcolor="#FFCCCC">
                <p align="center">
                    <a title="Ders Hakkında Duyurulara Bursan Ulaşabilirsiniz..." target="_blank"
href="duyuru.htm">
                        </a></td>
</tr>
<tr>
    <td width="133" height="43" bgcolor="#0083C1">
        <p align="center">
            <a target="I1" href="link.htm">
                </a></td>
            <td width="149" height="129" rowspan="3">
<form method="GET"
action="http://webmasterim.com/arama_motoru/arama_motoru.php"
target="arama_motoru">
<input type="hidden" name="s_i" value="http://ipekgoz.edu.tr.tc">
```

EK-1 (devam) Web sitesinin yapılmasında kullanılan bazı HTML kodları

```
<table border="0" cellpadding="0" cellspacing="0" style="border:2px dotted
#FF0000;; border-collapse:collapse" bordercolor="#111111">
<tr>
<td bgcolor="#004261" align="center">
<font face="Verdana" size="2" color="#FFFFFF"><b>Arama
Motoru</b></font></td>
</tr>
<tr>
<td bgcolor="#004261" width="140" height="60" align="center">
<font face="Verdana" size="1" color="#FFFFFF"><b>Web'de Ara:</b></font><br>
<input type="text" name="q" size="18" style="font-size: 8pt; background-color:
#f5f5f5; font-weight: bold; font-family: verdana;">
</td>
</tr>
<tr>
<td bgcolor="#004261" align="center">
<input type="submit" value=" Ara " style="font-size: 8pt; background-color:
#f5f5f5; font-weight: bold; font-family: verdana;" onClick="openw();">
</td>
</tr>
</table>
</form>
</td>
</tr>
<tr>
<td width="133" height="43" bgcolor="#0083C1">
<p align="center">
<a href="ziyaretc/index.php" target="_blank">
</a></td>
```

EK-1 (devam) Web sitesinin yapılmasında kullanılan bazı HTML kodları

```

</tr>
  <tr>
    <td width="133" height="43" bgcolor="#0083C1">
      <p align="center">
        </td>
      </tr>
    <tr>
      <td width="133" height="17" bgcolor="#0083C1">
        <p align="center">
          <a href="http://www.statcounter.com/" target="_blank"></a></td>
        <td width="149" height="23" bgcolor="#006697">

</td>
      </tr>
    <tr>
      <td width="133" height="1" bgcolor="#0083C1">
        &nbsp;</td>
      <td width="149" height="28" bgcolor="#0083C1">
&nbsp;</td>
    </tr>
  </table>
</center>
</div>
</body>
</html>

```


EK-1 (devam) Web sitesinin yapılmasında kullanılan bazı HTML kodları

```
<p class="MsoNormal" align="justify">
<span style="font-weight: 700; font-family: Verdana">
<font size="2" color="#FFFFFF">Burada bizim amacımız komple bir dersin
işlenişini her yönüyle ve verimli olacak şekilde internet ortamında uygulamaya
çalışmaktır. Bu durumda öğrencilerimiz internet üzerinden
istedikleri zaman ve istedikleri yerde ders çalışma imkanı bulmuş
olacaklardır.Buda verilen dersin başarı düzeyini artıracaktır. </font></span></p>
<p class="MsoNormal" align="left">
&nbsp;</p>
<p class="MsoNormal" align="left">&nbsp;</p>
<p class="MsoNormal" align="left">
&nbsp;</p>
<p class="MsoNormal" align="left">&nbsp;</p>
<p class="MsoNormal" align="left">&nbsp;</p>
<p class="MsoNormal" align="left">&nbsp;</p>
<p class="MsoNormal" align="left">&nbsp;</p>
<p class="MsoNormal" align="left">&nbsp;</p>
</body>
</html>
```

Vizyon.htm

```
<html>
<head>
<meta name="GENERATOR" content="Microsoft FrontPage 5.0">
<meta name="ProgId" content="FrontPage.Editor.Document">
<meta http-equiv="Content-Type" content="text/html; charset=windows-1254">
<title>Yeni Sayfa 1</title>
</head>
```

EK-1 (devam) Web sitesinin yapılmasında kullanılan bazı HTML kodları

```
<body bgcolor="#000080">
<p align="center">
<font color="#FFFF00" face="Verdana" size="3">
    <span style="font-weight: 700; background-color: #0000FF">GENEL
BİLGİ</span></font></p>
<p align="justify">
<span style="font-weight: 700; font-family: Verdana">
<font size="2" color="#FFFFFF">Her geçen gün daha fazla üniversite internet'e
bağlanmakta ve derslerini web üzerinde geliştirebilme imkanı sayesinde akademik
programlarını geliştirme ve yaygınlaştırma alanında başarılı
olabilmektedirler.</font></span></p>
<p class="MsoNormal" align="justify">
<span style="font-weight: 700; font-family: Verdana">
<font size="2" color="#FFFFFF">Günümüzde eğitimciler bilgi ve becerilerini
geleneksel derslik eğitimlerinin dışında sınırlar ötesindeki öğrencilere
taşılmaktadırlar. Elektronik-öğrenim yoluyla tüm dünyada çeşitli ülkeler,
dinler ve farklı insan topluluklarına eşit fırsatlar sunarak birleştirici bir
rol oynamaktadırlar.Küresel köyün eğitim toplumları ömür boyu sürecek bir
dostluk&nbsp; ve arkadaşlıkla birbirlerine kenetlenmektedir.</font></span></p>
<p class="MsoNormal" align="center">
</p>
<p class="MsoNormal" align="justify">
<span style="font-weight: 700; font-family: Verdana">
<font size="2" color="#FFFFFF">Burada bizim amacımız komple bir dersin
işlenişini her yönüyle ve verimli olacak şekilde internet ortamında uygulamaya
çalışmaktır. Bu durumda öğrencilerimiz internet üzerinden
istedikleri zaman ve istedikleri yerde ders çalışma imkanı bulmuş
olacaklardır.Buda verilen dersin başarı düzeyini artıracaktır. </font></span></p>
<p class="MsoNormal" align="left">
&nbsp;</p>
```

EK-1 (devam) Web sitesinin yapılmasında kullanılan bazı HTML kodları

```
<p class="MsoNormal" align="left">&nbsp;</p>
<p class="MsoNormal" align="left">
&nbsp;</p>
<p class="MsoNormal" align="left">&nbsp;</p>
<p class="MsoNormal" align="left">&nbsp;</p>
<p class="MsoNormal" align="left">&nbsp;</p>
<p class="MsoNormal" align="left">&nbsp;</p>
<p class="MsoNormal" align="left">&nbsp;</p>
<p class="MsoNormal" align="left">&nbsp;</p>
</body>
</html>
```

Diornekler.htm:

```
<html>
<head>
<meta name="GENERATOR" content="Microsoft FrontPage 5.0">
<meta name="ProgId" content="FrontPage.Editor.Document">
<meta http-equiv="Content-Type" content="text/html; charset=windows-1254">
<title>Yeni Sayfa 2</title>
</head>
<body bgcolor="#000080">
<div align="center">
<center>
<table border="0" cellpadding="0" cellspacing="0" style="border-collapse:
collapse" bordercolor="#111111" width="98%" id="AutoNumber2"
bgcolor="#0000FF">
<tr>
<td width="100%" align="center" bgcolor="#0000FF">
```


EK-2 Web sitesinin yapılmasında kullanılan bazı PHP kodları

Anket.php

```
<?php
include "func.inc.php";
$v=new veritabani();
$v->sql="SELECT * FROM anket where aktif='1'";
$v->sorgula();
$sonuc=$v->cek();
$soru=$sonuc["soru"];
$id=$sonuc["id"];
$v->sql="SELECT * FROM anketcevap where anket='$id'";
$v->sorgula();
?>

<form action="anketsonuc.php" method="post" name="anketform" id="anketform">
  <table width="200" border="0" cellspacing="0" cellpadding="0">
    <tr>
      <td><?php echo $soru; ?></td>
    </tr>
  </table>
  <?php
  $tik=0;
  while ($sonuc=$v->cek()){
    $cevap=$sonuc["cevap"];
    $aid=$sonuc["id"];
    $tik+=$sonuc["tik"];
  }
  ?>

  <tr>
    <td><input name="oy" type="radio" value="<?php echo $aid; ?>"> <?php echo
    $cevap; ?></td>
  </tr>
</table>
<?php
```

EK-2 (devam) Web sitesinin yapılmasında kullanılan bazı PHP kodları

```

}
?>
<tr>
    <td><?php echo $tik; ?> ki&#351;i anketi cevaplad&#305;.</td>
</tr>
<tr>
    <td>
        <input type="submit" name="Submit" value="Oy Ver">
        <input name="id" type="hidden" value="<?php echo $id; ?>"></td>
    </tr>
</table>
</form>

```

Anketgir.php

```

<?php
session_start();
if(! session_is_registered("login")) $login=0;
session_register("login");
if ($user=@$_POST["user"]){
    $pass=@$_POST["pass"];
    if ($user="admin" && $pass="1"){
        echo "<meta http-equiv=\"refresh\"
content=\"0;url=anketgir.php\">Bekleyin...";
        $login=1;
        exit();
    }
}
if ($login==0){
    echo "Kullanıcı adı veya şifre hatalı...";
}

```

EK-2 (devam) Web sitesinin yapılmasında kullanılan bazı PHP kodları

```

exit;
}
include "func.inc.php";
$v=new veritabani;
$v->sql="Select * From anket ORDER BY `id` DESC;";
$asno=$v->sorgula();
?>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<title>AnketGirişi</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-9">
</head>
<body>
<form name="form1" method="post" action="anketislem.php">
  <table width="600" border="0" align="center" cellpadding="0" cellspacing="0"
bgcolor="#CCCCCC">
    <tr>
      <td width="146">
        <div align="center">
          <select name="islemtip" id="islemtip">
            <option value="2">Cevap</option>
            <option value="1">Anket</option>
          </select>
        </div></td>
      <td width="370"><input name="deger" type="text" id="deger" size="60"></td>
      <td width="84"><div align="center">
        <input type="submit" name="Submit" value="Ekle">
      </div></td>
    </tr>
  </table>
</form>

```

EK-2 (devam) Web sitesinin yapılmasında kullanılan bazı PHP kodları

```

</tr>
</table>
<br>
<table width="600" border="1" align="center" cellpadding="0" cellspacing="0"
bordercolor="#000000" bgcolor="#CCCCCC">
<?php
$i=0;
while ($sonuc=$v->cek($asno)){
    $id=$sonuc[0];
    $soru=$sonuc[1];
    $aktif=$sonuc[2];
?>
<tr>
<td width="517"><input name="anket" type="radio" value="<?php echo $id;
?>" <?php
    if ($i==0){
        $i=1;
        echo " checked ";
    }
?>
>
<?php echo $soru;
    if ($aktif==1) echo " (Aktif)";
    else echo " (Pasif)";?></td>
<td width="41"><div align="center"><a
href="anketislem.php?islem=1&id=<?php echo $id; ?>">Sil</a></div></td>
<td width="42"><div align="center"><a
href="anketislem.php?islem=3&id=<?php echo $id; ?>">Aktif</a></div></td>
</tr>
<?php

```

EK-2 (devam) Web sitesinin yapılmasında kullanılan bazı PHP kodları

```
$v->sql="Select * From anketcevap Where anket='$id';";
    $v->sorgula();
    while ($sonuc=$v->cek()){
        $cid=$sonuc[0];
        $cevap=$sonuc[2];
        $tik=$sonuc[3];
    ?>
    <tr>
        <td>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;<?php echo $cevap." ( ".$tik." )";
    ?></td>
    <td colspan="2"><div align="center"><a href="anketislem.php?islem=2&id=<?php
echo $cid; ?>">Sil</a></div></td>
    </tr>
<?php
    }
}
?>
</table>
<p>&nbsp;&nbsp;&nbsp;</p>
</form>
</body>
</html>
```

Anketislem.php

```
<?php
session_start();
if(! session_is_registered("login")) $login=0;
session_register("login");
```

EK-2 (devam) Web sitesinin yapılmasında kullanılan bazı PHP kodları

```

if ($login==0){
    echo "Yetkiniz yok...";
    exit;
}
include "func.inc.php";
$v=new veritabani;
if ($islem=@$_GET["islem"]){
    $id=@$_GET["id"];
    if ($islem==1){
        $v->sql="DELETE FROM anket WHERE id='$id'";
        $v->sorgula();
        $v->sql="DELETE FROM anketcevap WHERE anket='$id'";
        $v->sorgula();
    }elseif ($islem==2){
        $v->sql="DELETE FROM anketcevap WHERE id='$id'";
        $v->sorgula();
    }elseif ($islem==3){
        $v->sql="UPDATE `anket` SET `aktif` = '0'";
        $v->sorgula();
        $v->sql="UPDATE `anket` SET `aktif` = '1' Where id='$id'";
        $v->sorgula();
    }
}
}else{
    if ($islemtip=@$_POST["islemtip"]){
        $anket=@$_POST["anket"];
        $deger=@$_POST["deger"];
        if ($islemtip==1){
            $v->sql="INSERT INTO `anket` (`soru`) VALUES ('$deger')";
            $v->sorgula();
        }
    }
}

```

EK-2 (devam) Web sitesinin yapılmasında kullanılan bazı PHP kodları

```

        if ($anket==0){
            echo "hatali giris yaptin";
            exit();
        }
        $v->sql="INSERT INTO `anketcevap` (`anket`,`cevap`) VALUES
('$anket','$deger')";
        $v->sorgula();
    }
}
}

```

```

echo "<meta http-equiv=\"refresh\" content=\"0;url=anketgir.php\">Bekleyin...";
?>

```

Anketlogin.php

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<title>Untitled Document</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
</head>
<body>
<p>&nbsp;</p>
<p>&nbsp;</p>
<form name="form1" method="post" action="anketgir.php">
    <table width="400" border="0" align="center" cellpadding="0" cellspacing="0"
bgcolor="#CCCCCC">
        <tr>

```

EK-2 (devam) Web sitesinin yapılmasında kullanılan bazı PHP kodları

```

<td>&nbsp;</td>
    <td>&nbsp;</td>
</tr>
<tr>
    <td width="175"><div align="right">Kullanici : </div></td>
    <td width="225"><input name="user" type="text" id="user"></td>
</tr>
<tr>
    <td><div align="right">&#350;fre : </div></td>
    <td><input name="pass" type="password" id="pass"></td>
</tr>
<tr>
    <td>&nbsp;</td>
    <td><input type="submit" name="Submit" value="Tamam"></td>
</tr>
<tr>
    <td>&nbsp;</td>
    <td>&nbsp;</td>
</tr>
</table>
</form>
<p>&nbsp;</p>
</body>
</html>

```

Anketsonuc.php

```

<?php
include "func.inc.php";
$v=new veritabani;

```

EK-2 (devam) Web sitesinin yapılmasında kullanılan bazı PHP kodları

```

if ($oy=@$_POST["oy"]){
    $v->sql="Select tik FROM anketcevap WHERE id='$oy'";
    $v->sorgula();
    $sonuc=$v->cek();
    $tik=$sonuc["tik"]+1;
    $v->sql="UPDATE `anketcevap` SET `tik` = '$tik' WHERE `id` ='$oy'";
    $v->sorgula();
}
$v->sql="SELECT * FROM anket where aktif='1'";
$v->sorgula();
$sonuc=$v->cek();
$soru=$sonuc["soru"];
$id=$sonuc["id"];
$v->sql="SELECT * FROM anketcevap where anket='$id'";
$v->sorgula();
?>
<table width="200" border="0" cellspacing="0" cellpadding="0">
    <tr>
        <td><?php echo $soru; ?></td>
    </tr>
</table>
<?php
$ttik=0;
while ($sonuc=$v->cek()){
    $cevap=$sonuc["cevap"];
    $aid=$sonuc["id"];
    $tik=$sonuc["tik"];
    $ttik+=$tik;
}
?>
<tr>
    <td> <?php echo $cevap." (".$tik.)"; ?></td>

```

EK-2 (devam) Web sitesinin yapılmasında kullanılan bazı PHP kodları

```

    </tr>
<?php
}
?>
<tr>
    <td><?php echo $ttik; ?> ki&#351;i anketi cevaplad&#305;.</td>
</tr>
<tr>
    <td>
        <input name="id" type="hidden" value="<?php echo $id; ?>"></td>
    </tr>
</table>

```

Creatanketdb.php

```

<?php
include "func.inc.php";
$v=new veritabani;
$v->sql="CREATE TABLE `anket` (
    `id` int( 6 ) unsigned NOT NULL AUTO_INCREMENT ,
    `soru` varchar( 60 ) binary NOT NULL default '',
    `aktif` int( 1 ) unsigned NOT NULL default '0',
    PRIMARY KEY ( `id` )
) TYPE = MYISAM ;
CREATE TABLE `anketcevap` (
    `id` int( 9 ) unsigned NOT NULL AUTO_INCREMENT ,
    `anket` int( 6 ) unsigned NOT NULL default '0',
    `cevap` varchar( 100 ) binary NOT NULL default '',

```

EK-2 (devam) Web sitesinin yapılmasında kullanılan bazı PHP kodları

```
PRIMARY KEY ( `id` ); TYPE = MYISAM ;";
```

```
$v->sorgula();
```

```
?>
```

Anketle ilgili SQL (Veri tabanı)

```
CREATE TABLE `anket` (
```

```
`id` int( 6 ) unsigned NOT NULL AUTO_INCREMENT ,
```

```
`soru` varchar( 60 ) binary NOT NULL default "",
```

```
`aktif` int( 1 ) unsigned NOT NULL default '0',
```

```
PRIMARY KEY ( `id` )
```

```
) TYPE = MYISAM ;
```

```
CREATE TABLE `anketcevap` (
```

```
`id` int( 9 ) unsigned NOT NULL AUTO_INCREMENT ,
```

```
`anket` int( 6 ) unsigned NOT NULL default '0',
```

```
`cevap` varchar( 100 ) binary NOT NULL default "",
```

```
`tik` int( 6 ) unsigned NOT NULL default '0',
```

```
PRIMARY KEY ( `id` )
```

```
) TYPE = MYISAM
```

ÖZGEÇMİŞ

Kişisel Bilgiler

Soyadı, adı : PEKGÖZ, İsmail
Uyruğu : T.C.
Doğum tarihi ve yeri : 25.12.1969 Amasya
Medeni hali : Evli
Telefon : 0 (358) 218 70 07
Cep(Tel.) : 0 (536) 464 07 34
e-mail : ipekgoz@hotmail.com

Eğitim

Derece

Lisans

Eğitim Birimi

Gazi Üniversitesi

Elektronik-Bilg.Eğt.

Mezuniyet tarihi

1993

Lise

Amasya Lisesi

1987

İş Deneyimi

Yıl

1994-2006

Yer

Ondokuz Mayıs Üniversitesi

Görev

Öğretim Görevlisi

Yabancı Dil

İngilizce

Hobiler

Bilgisayar teknolojileri, Futbol,Satranç