MARMARA ÜNİVERSİTESİ YAYINEVİ

T.C.
MARMARA ÜNİVERSİTESİ
İSTATİSTİK ve KANTİTATİF ARAŞTIRMALAR
ve UYGULAMA MERKEZİ
İSKAR
Yayın No.98/1

SİSTEM ANALİZİ

C PROGRAMLAMA DİLİ ÖRNEKLERİYLE

Dr. Bülent AYYILDIZ

MARMARA ÜNİVERSİTESİ

QA
76.9
A99
1998

İSTANBUL, 1998

MARMARA ÜNİVERSİTESİ YAYINEVİ

LIBRARY UNIVERSITY

Prof. Dr. Kenan T.K. Sarafından
Çankaya Üniversitesi Kütüphanesi'ne
Bağışlanmıştır

MARMARA ÜNİVERSİTESİ

Çankaya University Library

MARMARA ÜNİVERSİTESİ YAYINEVİ

B015271

MARMARA ÜNİVERSİTESİ YAYINEVİ

İSKAR
T.C.
MARMARA ÜNİVERSİTESİ
İSTATİSTİK ve KANTİTATİF ARAŞTIRMALAR
ve UYGULAMA MERKEZİ
Yayın No.98/1

SİSTEM ANALİZİ

C PROGRAMLAMA DİLİ ÖRNEKLERİYLE

Dr. Bülent AYYILDIZ

Çankaya University Library

İSTANBUL, 1998

MARMARA ÜNİVERSİTESİ YAYINEVİ

QA
76.9
A99
1998

MARMARA ÜNİVERSİTESİ YAYINEVİ

İÇİNDEKİLER

1. GİRİŞ 1
2. SİSTEM KAVRAMI 2
2.1. SİSTEM TANIMI 2
2.2. SİSTEMLERİN SINIFLANDIRILMASI 2
2.3. SİSTEM, ALT-SİSTEM İLİŞKİSİ VE İŞLETME SİSTEMİ 3
3. SİSTEM MODELLERİ 6
3.1. MODELİN TANIMI 6
3.2. MODEL KURMA AŞAMALARI 6
3.3. MODELLERİN SINIFLANDIRILMASI 7
3.3.1. FONKSİYONLARINA GÖRE MODELLER 8
3.3.2. YAPILARINA GÖRE MODELLER 8
3.3.3. TEMSİL ŞEKİLLERİNE GÖRE MODELLER 9
3.3.4. ZAMAN ESASINA GÖRE MODELLER 9
3.3.5. EDİNİLEN BİLGİYE GÖRE MODELLER 10
3.3.6. KAPSAMLARINA GÖRE MODELLER 11
4. SİSTEM ANALİZİ 11
5. SİSTEM YAKLAŞIMI 12
5.1. SİSTEM YAKLAŞIMININ AŞAMALARI 12
5.2. SİSTEM YAKLAŞIMINDA KULLANILAN ARAÇLAR 16
5.2.1. SİSTEM AKIŞ DİYAGRAMI 16
5.2.1.1. SİSTEM AKIŞ DİYAGRAMI SEMBOLLERİ 17
5.2.1.2. SİSTEM AKIŞ DİYAGRAMI ÖRNEKLERİ 20
5.2.2. PROGRAM AKIŞ DİYAGRAMI 23
5.2.2.1. PROGRAM AKIŞ DİYAGRAMI SEMBOLLERİ 23
5.2.2.2. PROGRAM AKIŞ DİYAGRAMI VE C PROGRAMLAMA DİLİ ÖRNEKLERİ 24
5.2.3. KARAR TABLOLARI 48
6. C PROGRAMLAMA DİLİ İLE BİR DOSYALAMA VE İSTATİSTİKSEL ANALİZ UYGULAMASI 54
7. SONUÇ 68
8. YARARLANILAN KAYNAKLAR 70

MARMARA ÜNİVERSİTESİ YAYINEVİ

1. GİRİŞ

İnsanoğlu evrensel bir sistem içinde doğar ve yaşamını sürdürür. Evrenin kendisi de bir sistemdir ve sonsuz sayıda alt sistemden oluşur. İnsan, kendisinin de bir sistem ve alt sistemler bütünü olarak yer aldığı evrensel sistem içinde, var olan tüm sistemleri anlamaya, açıklamaya ve kontrol etmeye çalışmaktadır. İnsanın sistemler ile ilgili olarak yaptığı bu çalışmalar ve çabalar genelde kendi sınırsız ihtiyaçlarının karşılanmasına yöneliktir. İnsanoğlu kendi ihtiyaçlarının karşılanmasında her türlü doğal sistemi kullanmaktadır. Ayrıca, yine ihtiyaçların karşılanması amacına yönelik olarak, doğal sistemleri taklit eden yapay sistemler üretmekte ve kullanmaktadır.

Bu çerçeve içinde, çalışmamın ikinci bölümünde sistem kavramı ele alınarak, tanımı, sistemlerin sınıflandırılması ve sistem, alt-sistem ilişkisine bağlı olarak işletme sistemi incelenmiştir.

İnsanoğlu, sistemleri anlamak amacıyla yaptığı çalışmalar sırasında, kompleks yapıları nedeniyle bir çok sistemin çalışmasını anlamakta güçlük çekmiştir. Bu aşamada gerçek sistemi, belirli ölçekte temsil eden modeller kullanılmıştır. Çalışmamın üçüncü bölümünde, sistem modelleri; tanımı, model kurma aşamaları ve modellerin sınıflandırılması itibariyle incelenmiştir.

Dördüncü bölümde sistem analizi incelenerek, çağdaş sistem analizinin gelişmesine etki eden faktörler ortaya konmuştur.

Beşinci bölümde ise sistem yaklaşımı ele alınmıştır. Sistem analizini de kapsadığı düşüncesi ile sistem yaklaşımı; aşamaları ve sistem yaklaşımında kullanılan araçlar itibariyle incelenmiştir. Sistem yaklaşımının aşamaları; mevcut sistemin analizi, yeni sistemin tasarımı, tasarlanan sistemin kurulması ve kurulan yeni sistemin işletilmesi ve izlenmesi olarak belirlendikten sonra, bu aşamalarda izlenmesi gereken yaklaşımlar sorulan sorular ile belirlenmeye çalışılmış ve çeşitli aşamalarda kullanılabilecek yöntemler ve araçlar sıralanmıştır.

Sistem yaklaşımında kullanılan araçlar beşinci bölümde sıralandıktan sonra, yine bu bölüm içinde sistem akış diyagramları, program akış diyagramları ve karar tabloları tek, tek ele alınarak, incelenmiştir. Sistem analizinin önemli araçları olan sistem akış diyagramları, program akış diyagramları ve karar tabloları çeşitli örnekler sunularak canlandırılmaya ve kullanımı konularında çeşitli bakış açıları sunulmaya çalışılmıştır. Ayrıca, program akış diyagramları ve karar tablolarına bağlı olarak sunulan C programları bu araçların kullanımını daha anlamlı ve yararlı hale getirmiştir.

Çankaya University Library

MARMARA ÜNİVERSİTESİ YAYINEVİ

MARMARA ÜNİVERSİTESİ YAYINEVİ

2. SİSTEM KAVRAMI

Günlük yaşamımızda sistem kelimesi çok sık kullanılan bir kelimedir. Bilişim sisteminin kurulması, eğitim sisteminin reorganizasyonu, üretim sisteminin verimliliği, savunma sistemi yatırımları insanlar tarafından çok sık konuşulan, yazılan, çözümlenmek istenen konulardır. Burada ortak sözcük sistemdir. Bu şekilde yaklaşıldığında her şey bir sistemdir. Örneğin, bir insan, bir bitki, bir şehir, bir işletme, bir okul hep birer sistemdir.

2.1. SİSTEM TANIMI

Sistemin çeşitli tanımları yapılmıştır. Bunlardan bazıları şu şekilde sıralanabilir.

Sistem, bir veya daha çok amaca veya sonuca,ulaşmak üzere,aralarında ilişkiler olan fiziksel veya kavramsal, birden çok bileşenin oluşturduğu bütündür. [1]

Bir amaca ulaşmak için birlikte faaliyet gösteren bileşenler kümesidir.[2]

Schoderbek ve Kefalas'a göre; sistem, bir bütünlük ve tamlık oluşturmak üzere elemanlarının birbirleriyle ve çevreleriyle ilişkide olduğu ve belirli bir amaca yönelik bir elemanlar kümesidir.

Sistem kavramının tanımı çeşitli kaynaklarda yapılmıştır. Genelde ortak oldukları temel esas, her sistemin gerçekleştirmek için faaliyette bulunduğu bir amacı olduğudur. Ayrıca sistem, elemanlardan daha doğru bir ifade ile bileşenlerden oluşur. Sistem bileşenleri, birbirleriyle ilişki ve etkileşim içinde sistemin amacı doğrultusunda faaliyette bulunurlar.

Özet olarak, sistem belirli bir amacı gerçekleştirmek için faaliyette bulunan, birbirleri ile etkileşim içindeki elemanlar kümesidir, şeklinde tanımlanabilir.

2.2. SİSTEMLERİN SINIFLANDIRILMASI

Sistemler niteliklerine göre çeşitli şekillerde sınıflandırılırlar.

- **Açık ve Kapalı Sistemler :**

Tüm canlı sistemler açık sitemlerdir. Bu sistemler muhakkak bir çevreye sahiptirler ve çevreleriyle etkileşim içindedirler.

Kendi dışında hiçbir sitemin mevcut olmadığı sistemler veya kendi dışında ki sistemlerden hiçbir şekilde etkilenmeyen sistemlere kapalı sistemler adı verilir.[3]

- **Canlı ve Cansız Sistemler :**

Doğum, ölüm ve çoğalma gibi biyolojik özelliklere sahip sistemlere canlı sistemler denir. Örnek olarak yaşamın başladığı hücreden, sosyal organizasyonlara kadar tüm organizasyonlar canlı sistemlere örnek olarak verilebilir.

---

[1] Esen, Öner, "İşletme Yönetiminde Sistem Yaklaşımı", İ.Ü.İşletme Fakültesi Yayın No:174, İstanbul, 1985, s.10.

[2] Davis, S. Williams, "An Introduction to Business Information Systems", West Publishing Company, Minneapolis-St.Paul, 1995, s.186.

[3] Esen, op.cit s.15.

MARMARA ÜNİVERSİTESİ YAYINEVİ

Bir canlılık olmamakla beraber, cansız sistemlere ilişkin bazı süreçleri tanımlamakta da doğum ve ölüm kavramları kullanılmaktadır. Ancak gerçek anlamda canlılık göstermeyen bu sistemlere cansız sistemler denir. [4]

- **Soyut ve Somut Sistemler :**

Tüm elemanları kavramlardan oluşan sistemlere soyut sistemler denir. Tüm soyut sistemler cansız sistemlerdir.

En az iki elemanı nesnelerden oluşan sistemlere somut sistemler denir. Bir başka ifade ile somut sistemler kavramsal ve fiziksel nesnelerden oluşabilir. Somut sistemler canlı veya cansız olabilir.[5]

- **Doğal ve İnsan Yapısı Sistemler :**

Doğal sistemler, doğal süreçler sonunda oluşan, varlığını sürdüren ve gelişen sistemlerdir. Örnek olarak, güneş sistemini, iklim sistemini gösterebiliriz.

Tamamen insan tarafından meydana getirilen sistemler yanında, her ne şekilde olursa olsun insan katkısı olan her türlü sistem insan yapısı sistemler olarak kabul edilirler. Bir otomobil, ekonomik sistem, haberleşme sistemi bu kategoriye örnek olarak verilebilir.[6]

- **Uyum Sağlayan ve Uyum Sağlayamayan Sistemler :**

Açık sistemler daima bir değişim ile karşı karşıya gelirler. Eğer sistem bu değişikliği içeri alır, kullanır ve kendini geliştirirse uyumlu (adaptive) bir sistemdir. Aksi takdirde uyum sağlayamayan bir sistem olarak nitelendirilir.

- **Dinamik ve Statik Sistemler :**

Çevresel tüm değişiklikleri algılayan ve o değişikliklere uyum sağlayan sistemlere dinamik sistemler adı verilir. Bu sistemlere bilgi gelir. Sistem bilgiyi alır ve değişimini gerçekleştirir. Bu niteliği göstermeyen sistemlere de statik sistem adı verilir.

2.3. SİSTEM, ALT-SİSTEM İLİŞKİSİ VE İŞLETME SİSTEMİ

Bir sistemin verileri başka ve daha büyük bir sistem tarafından sunuluyorsa, bu sistem bir alt-sistem olarak nitelendirilir.[7] Örneğin bir işletme, finansman, üretim, pazarlama vb. alt-sistemlerden oluşur. İşletme sistemi ise, ülke ekonomik sisteminin bir alt-sistemidir. Ülke ekonomik sistemi ise, dünya ekonomik sisteminin bir alt-sistemidir. Genelde sistemler daha büyük bir sistemin alt-sistemidirler.

Dinamik bir sistem olan ekonomik sistemin temel elemanları şunlardır:

- Üreticiler
- Tüketiciler
- Sendikalar
- Bilimsel ve Teknolojik Çevre

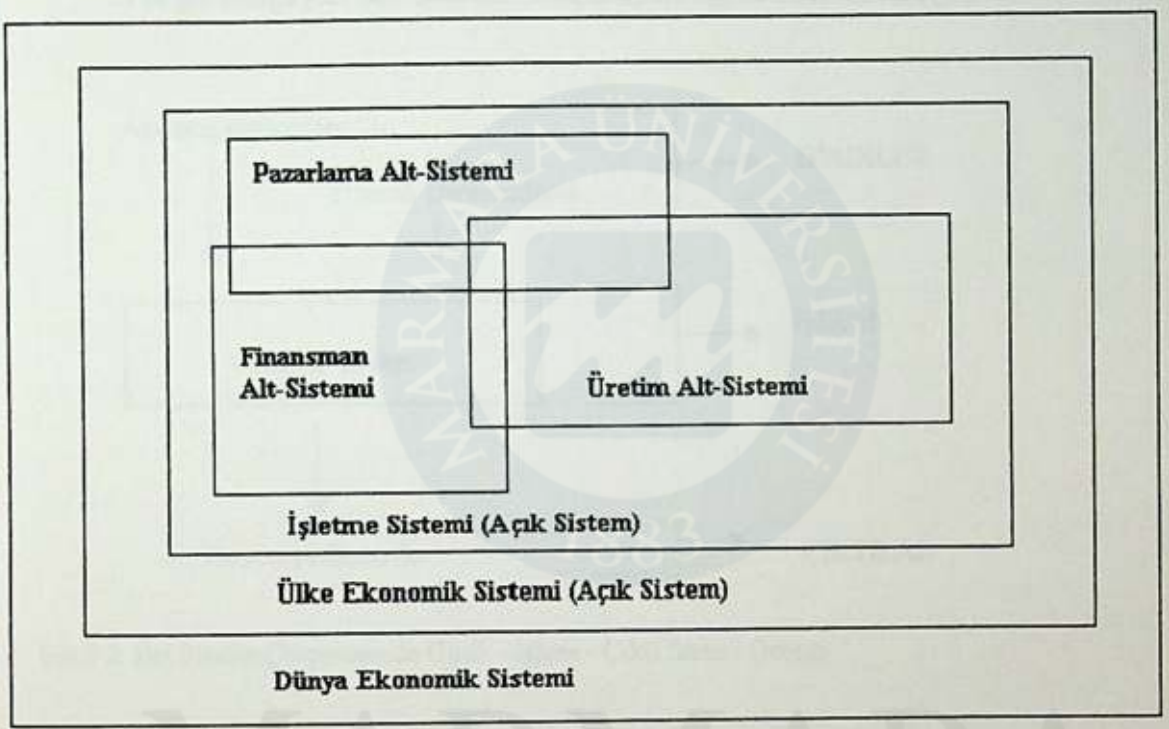
---

[4] Ibid
[5] Ibid s.16.
[6] Ibid
[7] İlyasoğlu, Eyüp, "Üretim Sistemlerinin Yönetimi", İstanbul, 1982, s.14.

MARMARA ÜNİVERSİTESİ YAYINEVİ

- Devlet ve Yasal Kurumlar
- Sosyal Çevre
- Yerel Kuruluşlar (belediyeler, hastaneler)

Ülke ekonomik sisteminin bir alt sistemi olan işletme temel amacı kâr olan bir üretim stemidir. İşletme mal veya hizmet üretimi yaparak en fazla kârı elde etmeye çalışır. Şekil 'de işletme sistemi, sistem-alt sistem ilişkisi içinde verilmiştir.



Şekil 1. Sistem-Alt Sistem İlişkisi

şletme Sisteminin Girdileri :

- Teşebbüs Bilgileri
- Hammaddeler
- Teknik Bilgiler
- Makine
- İşgücü
- Toplumsal Amaçlar
- Bireysel Ekonomik İstekler
- Diğer Girdiler

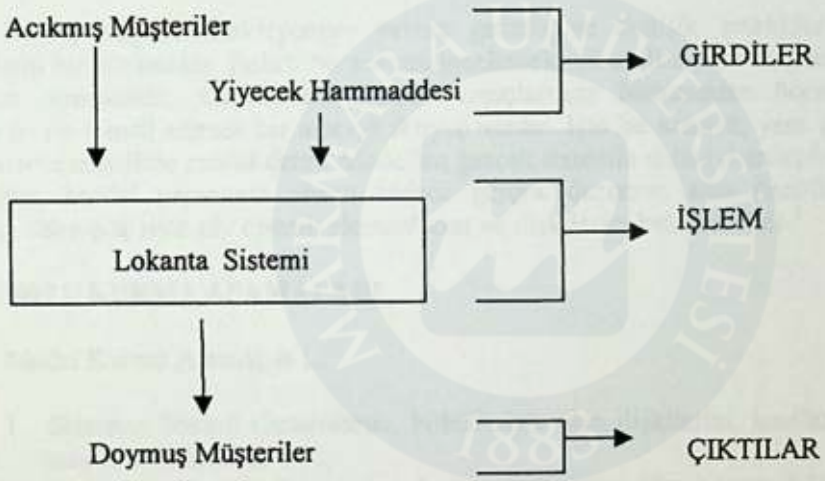
İşletme Sisteminin Çıktıları :

- Mamuller
- Hizmetler

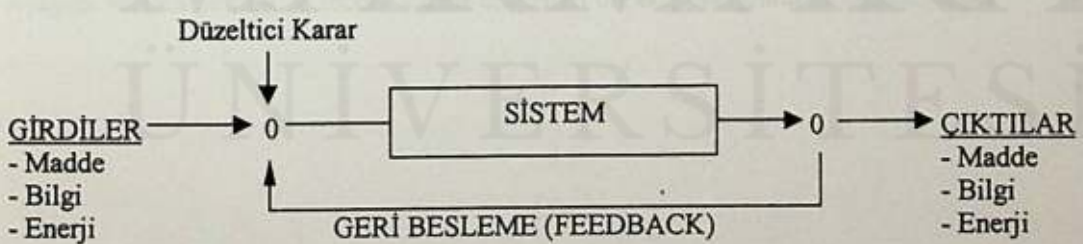
MARMARA ÜNİVERSİTESİ YAYINEVİ

- Negatif Çıktılar
- Diğer Çıktılar

Şekil 2'de verilen örnekte görüldüğü gibi, sistemlerin girdileri ve çıktıları vardır. Sistem içinde işlemden geçen girdiler, çıktı olarak elde edilir. İstenilen nitelikteki çıktıları elde edebilmek çoğu zaman mümkün olmayabilir. Çünkü sistem, alt-sistem ilişkisi içinde bir sistemin girdileri zaman içinde farklılık arz eder. Bu da çıktılardaki farklılığa neden olur. Çıktıların arzu edilen niteliğe sahip olması için sistemlerin Şekil 3'de görüldüğü gibi Geri Besleme (feedback) niteliği ile donatılmaları gerekir.



Şekil 2. Bir Sistem Çerçevesinde Girdi – İşlem - Çıktı Süreci Örneği



Şekil 3. Sistem ve Geri Besleme İlişkisi

MARMARA ÜNİVERSİTESİ 1883
MARMARA ÜNİVERSİTESİ

MARMARA ÜNİVERSİTESİ YAYINEVİ

3. SİSTEM MODELLERİ

Çoğu zaman sistem ve model kelimeleri birbirlerinin eş anlamlısı olarak kullanılırlar. Ancak gerçekte bu iki kavram birbirlerinden tamamen farklıdırlar.

- Sistem organize ve kompleks bir gerçektir.
- Model gerçeği temsil eder, ancak gerçek değildir ve model sistem kadar kompleks değildir.

3.1. MODELİN TANIMI

Evren, değişik fonksiyonları yerine getiren ve değişik niteliklere sahip farklı sistemlerin bir bütünüdür. Belirli bir sistemi incelemek, onun ilkelerini belirlemek her zaman mümkün olmayabilir. Gerçek bir sistemi amaçlarımız bakımından önem taşıyan tüm özellikleri ile temsil edecek bir aracıya ihtiyaç vardır. İşte bu aracıya, yani gerçek sistemin veya sürecin temsiline model denir. Modeller, gerçek sistemin daha iyi anlaşılmasını sağlayan aracılardır. Model yapmanın amacı sadece gerçek durumun tüm özelliklerinin temsili olmayıp, daha çok sistemin önemli elemanlarını ve ilişkilerini belirlemektir.[8]

3.2. MODEL KURMA AŞAMALARI

Model Kurma Aşamaları :

1. Sistemin önemli elemanlarını, birbirleriyle olan ilişkilerini, kısıtlarını ve sistemin amacını belirleme.
2. Birinci aşamada belirlenen temel sistem nitelikleri itibariyle modelin kurulması.
3. Modelin temsil ettiği sistem ile aynı amaca hizmet edip, edemeyeceğinin test edilmesi.
4. Eğer test sonucu olumsuzsa modelin gözden geçirilmesi ve üzerinde değişiklikler yapılarak yeniden test edilmesi.
5. Eğer test sonucu olumlu ise model uygulanması.

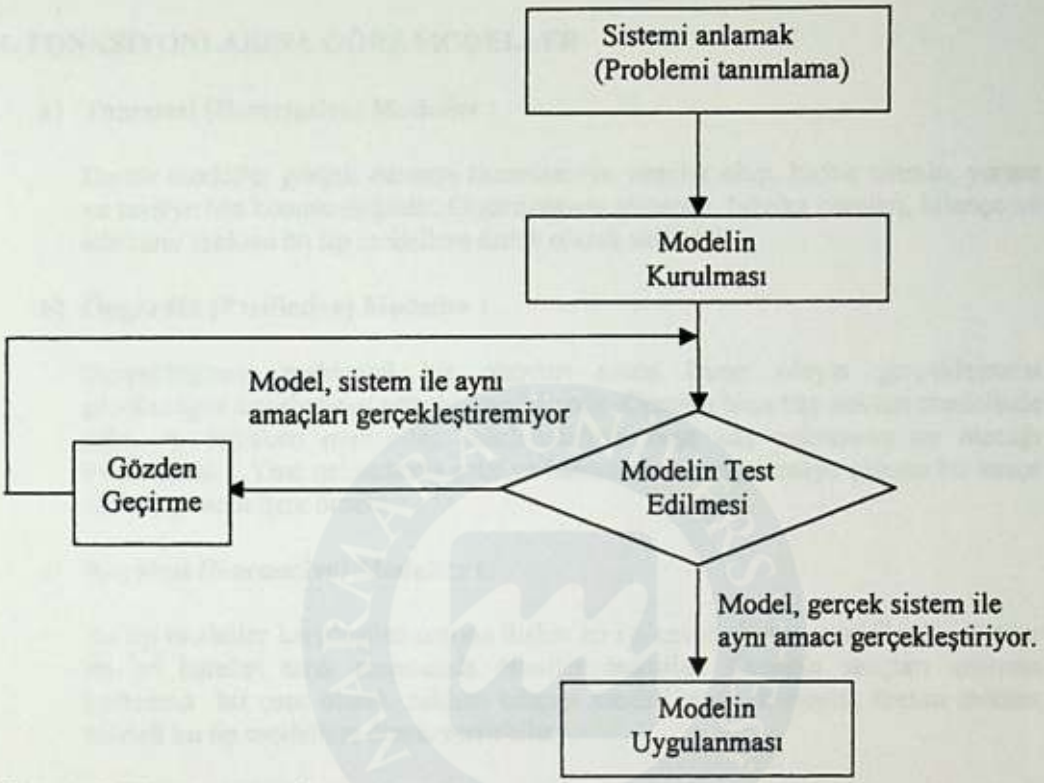
Aşağıda verilen Şekil 4'de model kurma aşamaları bir akış diyagramı ile açıklanmıştır.

[8] Esen, op.cit, s.18.

MARMARA ÜNİVERSİTESİ YAYINEVİ

MARMARA ÜNİVERSİTESİ 1883
MARMARA ÜNİVERSİTESİ

MARMARA ÜNİVERSİTESİ YAYINEVİ



Şekil 4. Model Kurma Aşamaları

Her modelde üç temel eleman vardır.

a. Amaç
b. Parametreler
c. Değişkenler

Basit Bir Üretim Modeli : $y = ax + b$

y : amaç
x : değişken
a : parametre
b : parametre

Parametreler belirli bir zaman dilimi içinde değişmeyen sabitlerdir.

3.3. MODELLERİN SINIFLANDIRILMASI[9]

Modeller ;

- fonksiyonlarına
- yapılarına
- temsil şekillerine,
- zaman esasına,
- elde edilen bilgiye ve
- kapsamlarına

göre sınıflandırmak mümkündür.

[9] Ibid s.22.

MARMARA ÜNİVERSİTESİ YAYINEVİ

MARMARA ÜNİVERSİTESİ YAYINEVİ

3.3.1. FONKSİYONLARINA GÖRE MODELLER

a) Tanımsal (Descriptive) Modeller :

Bu tür modeller gerçek durumu tanımlamaya yönelik olup, hiçbir tahmin, yorum ve tavsiye söz konusu değildir. Organizasyon şemaları, fabrika planları, bilanço ve kâr/zarar tablosu bu tip modellere örnek olarak verilebilir.

b) Öngörülü (Predictive) Modeller :

Gerçekleşmesi muhtemel bir olaydan sonra hangi olayın gerçekleşmesi gerekeceğini belirlemeye yönelik modellerdir. Örneğin başa baş noktası modelinde sabit ve değişken maliyetler belirlenebilirse başa baş noktasının ne olacağı belirlenebilir. Yine gelecekteki gelir ve harcamaları dengelemeye çalışan bir bütçe de bu tip modellere örnektir.

c) Kuralsal (Normative) Modeller :

Bu tip modeller karşılaşılan soruna ilişkin en iyi cevabı bulan modellerdir. Bunlar en iyi hareket tarzı konusunda öneriler getirirler. Örneğin satışları arttırma konusuna bir çare olarak, reklam bütçesi modeli veya ekonomik üretim miktarı modeli bu tip modellere örnek verilebilir.

3.3.2. YAPILARINA GÖRE MODELLER

Yapılarına göre modeller de üç başlık altında incelenebilir.

a) Uyuşum (Iconic) Modelleri :

Bu tür modeller temsil ettikleri sistemin bazı fiziksel özelliklerini ihtiva ederler. Bir fabrikanın planına uygun olarak yapılmış maketi, bir ev planı, bir araba modelinin prototipi bu modellere örnek olarak verilebilir.

b) Benzeşim (Analog) Modeller :

Modeli kuracak sistem veya süreçle bir paralellik kuracak şekilde gerçek duruma ilişkin bileşenlerin yerine geçebilecek (onların fonksiyonlarını ifa edecek) elemanlar desteğinde kurulan modellerdir. Örneğin borulardan oluşan bir su şebekesinde suyun akışı, elektrik telleriyle oluşturulan bir şebekede ki elektrik akımıyla temsil edilebilir. Bu şekilde oluşturulan modeller analog modellerdir.

c) Sembolik (Symbolic) veya Matematik Modeller :

Bu tip modeller gerçek sistem veya süreci çeşitli sembollerle tanımlayan modellerdir. Matematik modeller diğer modellere oranla daha çok işleme tabi tutulma, analiz olanağı sağlar.

$$KÂR = GELİR - MALİYET$$

$$Y = AX + B$$

MARMARA ÜNİVERSİTESİ YAYINEVİ

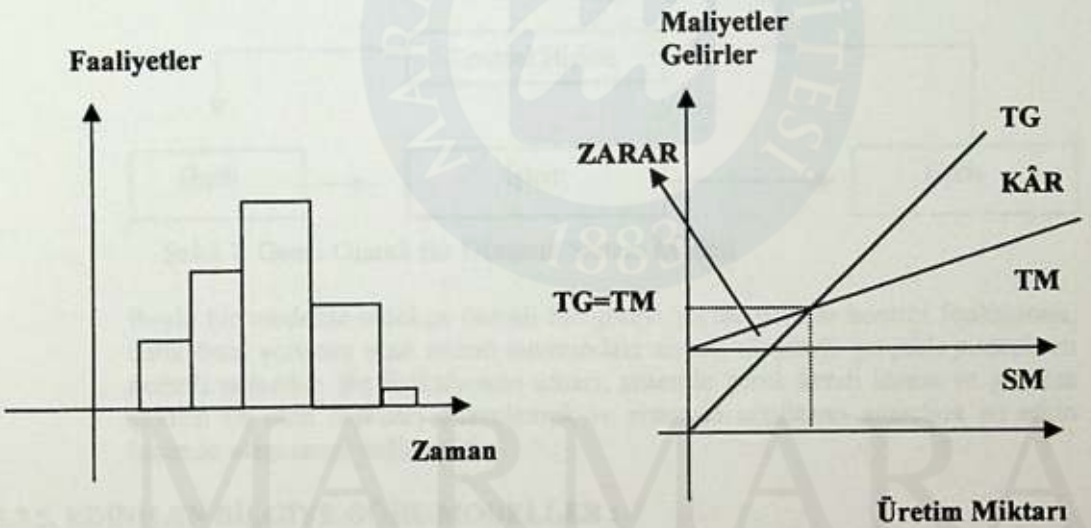
3.3.3. TEMSİL ŞEKİLLERİNE GÖRE MODELLER

a) Sözlü (Verbal) Modeller :

Bu tip modellerde gerçek durum veya sistem sözle anlatılır. Bu modellerde nesneler, ilişkiler ve süreçler kelimelerle tanımlanır. Örneğin "dünya, güneşin etrafında eliptik bir yörüngede dönmektedir." ifadesi sözlü bir modeldir.

b) Şematik (Schematic) Modeller :

Bu tip modellerde gerçek durum veya süreç resim veya sembollerle temsil edilir. Bu modeller, sistemin iç ve dış dinamiğini açık ve net olarak gösterirler. Histogramlar, grafikler, bu model tipine örnek olarak gösterilebilirler. Şekil 5 ve 6'da bu tür modellere örnek verilmiştir.



Şekil 5 –6. Şematik Model Örnekleri

3.3.4. ZAMAN ESASINA GÖRE MODELLER

a) Statik (Static) Modeller :

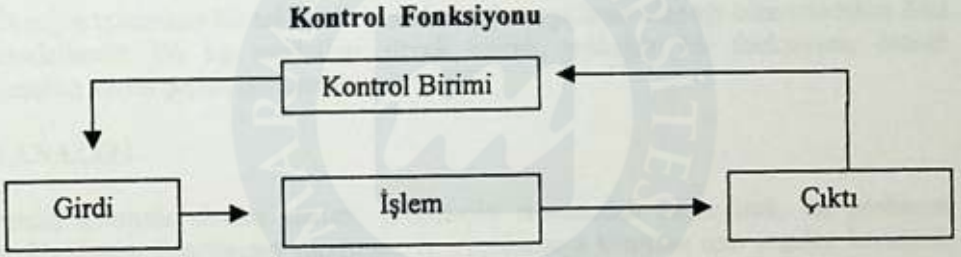
Bu tip modeller zaman boyutuna göre söz konusu olabilecek değişiklikleri dikkate almayan modellerdir. İşletmelerin bazı temel özelliklerini bu tip modellerle temsil etmek mümkündür. Örneğin organizasyon şemaları, zamana göre değişimleri

MARMARA ÜNİVERSİTESİ YAYINEVİ

içermeyen miktar, maliyet, kalite, yetki ve sorumlulukları içeren tüm modeller bu tip modellere örnek verilebilir.

b) Dinamik (Dynamic) Modeller :

Dinamik modellerin temel özelliği, belirli bir zaman boyutunda değişim gösterme yeteneğine sahip olmasıdır. Bu tip modeller, olayların meydana geldiği ortamda değişikliklerin gerçek yapısını ortaya koymaya yöneliktir. Dinamik modellerin temel ayırıcı özelliği kendi kendini düzenleyici bir dönüştürme sürecine sahip oluşudur. Dinamik modellerde girdi, dönüştürme süreci sonunda herhangi bir şekilde değişikliğe uğrayarak bir çıktıya dönüşmektedir. Bu tip modellere dinamik denmesinin en büyük nedeni, performanslarının kendileri tarafından ölçülmelerine ilişkin bir ölçü bulmaya yönelik olmasıdır. Şekil 7'de bir dinamik sistem modeli verilmiştir.



Şekil 7. Genel Olarak Bir Dinamik Sistem Modeli

Böyle bir modelde oldukça önemli bir görevi yerine getiren kontrol fonksiyonu, daha önce verilmiş olan sistem tanımındaki amaca ulaşmayı gerçekleştirecek en önemli unsurdur. Bu fonksiyonun amacı, sistemin gerek kendi içinde ve gerekse çevresi ile olan ilişkileri düzenlemek ve sistemin arzulanan amaçlara en etkin biçimde ulaşmasını sağlamaktır.

3.3.5. EDİNİLEN BİLGİYE GÖRE MODELLER :

a) Belirlilik Halinde (Deterministic) Modeller :

Modele ilişkin tüm değişken, parametre ve ilişkileri belli olan modellerdir. Bu tip modellere örnek olarak daha çok statik modeller verilebilir. Örneğin Kâr = Gelir-Maliyet tüm değişken ve ilişkileri belli olan bir modeldir.

b) Olasılık Taşıyan (Probabilistic, Under Risk) Modeller :

Bu tip modellerde girdi değerleri veya süreç, olasılık dağılımları ile temsil edilmektedir. Bu girdi değerlerinin olasılıklı oluşuna bağlı olarak çıktı değerleri de birden fazla olmakta hatta bir dağılım göstermektedir. Bu tip modellere örnek olarak yaşın bir fonksiyonu olarak, ölümleri veren tablolar veya Bayesian karar analizleri verilebilir.

MARMARA ÜNİVERSİTESİ YAYINEVİ

c) Belirsizlik Halinde (Under Uncertainty) Modeller :

Bu tip modeller, gerek modeli oluşturan bileşenler ve gerekse aralarındaki ilişkiler konusunda kesin veya olasılıksal hemen hiçbir bilgiye sahip olunamayan durumlarda optimum çözümler geliştirmenin yollarını arar. Oyun teorisi veya belirsizlik halinde karar teorisi modelleri bu tip modellere örnek olarak verilebilir.

3.3.6. KAPSAMLARINA GÖRE MODELLER

a) Genel Modeller :

Hemen her alanda uygulama alanı bulan modellerdir. Örnek olarak doğrusal programlama modelleri veya bekleme modelleri verilebilir.

b) Özel Modeller :

Örneğin işletmede bir tek fonksiyonel alanda uygulama olanağı olan modeller özel modellerdir. Bu tip modellere örnek olarak, reklamın bir fonksiyonu olarak satışları veren çeşitli denklemler verilebilir.

4. SİSTEM ANALİZİ

En geniş anlamda sistem analizi; problemin tanımından başlayarak, bu problemi çözecek yeni bir sistem tasarımının yapılması ve uygulamaya konması için yapılan kapsamlı çalışma demektir.

Sistem analizi, bir sistem ve onun işlediği çevre hakkındaki faktörleri toplamayı, düzenlemeyi ve değerlendirmeyi kapsar. Sistem analizinin amacı, daha iyi bir sistem tasarlama ve geliştirmeye bir temel oluşturmak üzere sistemin tüm yönlerinin değişik, çevresel faktörler açısından incelemektir.[10]

Çağdaş sistem analizinin gelişmesine etki eden faktörler şu şekilde sıralanabilir.

- Bilgisayarın gelişmesi ve uygulama alanlarının genişlemesi
- Sibernetik (Tüm hayvanlar ve makinelerle ilgili kontrol ve haberleşme teorisi)
- Bilişim Sistemi (Donanım, yazılım, insan ve prosedürlerden oluşan bir sistem olup, amacı doğru insana, doğru yer ve zamanda gereken bilgiyi sunmaktır. Bunun için örgüt faaliyetlerinden doğan veriler bilgiye dönüştürülerek bilgisayar ortamında organize edilerek saklanırlar.)
- Karar Teorisi (Bir karara ulaşmak için kullanılan varsayımlar, yöntemler ve teorilerden oluşan istatistik dalı)
- Faktör Analizi (Bir değişken kümesindeki içsel ilişkileri incelemeyi amaçlayan çok değişkenli analiz türü)
- Matematik

[10] Ibid, s.40.

MARMARA ÜNİVERSİTESİ YAYINEVİ

5. SİSTEM YAKLAŞIMI

Birçok kaynakta sistem analizi yaklaşımı, sistem yaklaşımı ve sistem mühendisliği yaklaşımı ile aynı anlamda kullanılmaktadır. Ama sistem analizi çalışmasını, sistem yaklaşımı içinde düşünmek gerekir. Bundan hareketle yine sistem analizinin, sistem yaklaşımının bir özel hali olduğu söylenebilir.[11]

5.1. SİSTEM YAKLAŞIMININ AŞAMALARI

Sistem yaklaşımı aşağıda verilen dört temel aşamadan oluşur.

- Mevcut Sistemin Analizi
- Yeni Sistemin Tasarımı
- Tasarlanan Sistemin Kurulması
- Kurulan Yeni Sisteminin İşletilmesi ve İzlenmesi

Bilgisayarlar ve bilgisayarlarda kullanılan paket programlar, sistem yaklaşımının çeşitli aşamalarında, yapılacak işlemler, bu işlem aşamalarında sorulması gereken sorular şu şekildedir. [12]

A) MEVCUT SİSTEMİN ANALİZİ - SİSTEM ANALİZİ

A.1. Problemin Tanımı veya Formülasyonu

- Problem nedir?
- Nasıl ortaya çıkmıştır?
- Bu problemin önemli olduğuna kim inanmaktadır ?
- Neden önemlidir ?
- Bu gerçekten incelenmesi gereken problem midir ?
- Para kazandıracak mıdır ?
- Herhangi bir başka problem ile karşılaşıldığında, bu problemi incelemenin görünür ve akılcı gerekçeleri var mıdır ?

Kullanılan Teknikler : Kritik İnceleme

A.2. Sistem Yaklaşımı Projenin Organizasyonu

- Sistem (proje) gurubu için en iyi kompozisyon hangisidir ?
- Proje ne zaman tamamlanacaktır ?
- Ön proje çizelgesi hazırlanmış mıdır ?
- Ayrıntılı faaliyetler belirlenip atanmış mıdır ?

Kullanılan Teknikler : CPM - Critical Path Method - Kritik Yol Yöntemi

A.3. Sistemin Tanımı

- İncelenen sistemin elemanları ve ilişkileri nelerdir ?
- Uygun alt-sistemlere en iyi şekilde nasıl bölünebilir ?
- Alt- sistemler arasında ki ilişkiler nelerdir ?
- Sistemin akış-blok diyagramı kurulmuş mudur ?

---

[11] Ibid, s.42.
[12] Erkut, Haluk, "Sistem Analizi", Kıyı Yayınları 802, İstanbul, 1989, s.43-58.

MARMARA ÜNİVERSİTESİ YAYINEVİ

A.4. Üst Sistemin Tanımı

- Sistemin içinde yer aldığı ve işletileceği çevre (ticari, sosyal, politik, vb. üst-sistem ve rakip sistemler)
- Sistemin, çevre ile ilişkileri nasıldır ?

A.5. Üst Sistemin Hedeflerinin Tanımı

- Üst düzey sistemler ile rakip sistemler yeterince hesaba katıldı mı ?
- Bunlar incelenen sistemin hedeflerini nasıl etkiliyor?

Kullanılan Teknikler : Stratejik Planlama

A.6. Sistemin Hedeflerinin Tanımı

- Sistemin hedefleri önem sırasına göre bir liste halinde düzenlendi mi ?
- Kısıtlar listelendi mi ?
- Hedefler basit ve açık mı ?
- Hedefler, sistem tasarımı ile ilgili herkese iletildi mi ?
- Hedefler üzerinde anlaşmaya varıldı mı ?

Kullanılan Teknikler : Amaçlara Göre Yönetim

A.7. Tüm Ekonomik Ölçütlerin Tanımı

- Çelişkili hedefler ağırlıklandırıldı mı ?
- Kısıtlar akılcı mı ?
- Ölçüt ve kısıtlar basit ve açık mı ?
- Nitel olsalar bile açık bir şekilde kabul edildiler mi ?

Kullanılan Teknikler : a. Finansal Planlama ve Ekonomik Modelleme
b. İndirgenmiş Nakit Akımı Yöntemi

A.8. İletişim ve Veri Toplama

- Tüm önemli kişiler ve kaynaklar ile görüşme yapıldı mı ?
- Tüm ilgili veriler toplanıp en iyi şekilde düzenlendi mi ?

Kullanılan Teknikler : İstatistik Veri Toplama Teknikleri

a. Anket Yöntemi
b. Gözlem Yöntemi
c. Deney Yöntemi
d. Projeksiyon Yöntemi

B. YENİ SİSTEMİN TASARIMI - SİSTEM TASARIMI

B.1. Tahmin

- Tahmin
- Sistemin çevresini öngörebilmek için tüm ilgili veriler kullanıldı mı ?
- Öngörüler ne derece doğrudur ?

MARMARA ÜNİVERSİTESİ YAYINEV

Kullanılan Teknikler :

a. İstatistik Tahmin Yöntemleri kullanılarak, kısa ve orta vadeli tahminler yapılır.
b. Teknolojik Tahmin.

B.2. Model Kurma

- Eldeki amaca en uygun model hangisidir ?
- Model kurma, en fazla gereksinim duyulan noktada yoğunlaştırılmış mıdır ?

Kullanılan Teknikler :

a. Ampirik Düzgün Durum Modelleri
b. Ampirik Dinamik Modeller
c. Mekanik Düzgün Durum Modelleri
d. Mekanik Dinamik Modeller
e. Ekonomik Modelleme
f. Nümerik Matematik
g. Monte Karlo Simülasyonu
h. Deterministik Simülasyon
i. Analog Bilgisayar Simülasyonu

B.3. Optimizasyon

- Hangi optimizasyon yöntemi kullanılmaktadır ?
- Performansa duyarlı değişkenler ayrıldı mı ?
- Ekonomik ölçütün, temel değişkenler türünden değişimi incelendi mi ?
- Tasarımın varsayımlara olan değişkenler türünden değişimi incelendi mi ?
- Tasarımın varsayımlara olan duyarlılığı denetlendi mi ?
- Risk analizi yapıldı mı ?

Kullanılan Teknikler :

a. Lineer Programlama
b. Non-Lineer Programlama
c. Matematik Programlama Yöntemleri
d. Risk Analizi
e. Karar Teorisi

B.4. Denetim

- Bir denetim sistemi kurarak, en uygun koşulların, en ekonomik olarak nasıl gerçekleşeceği araştırıldı mı?
- Denetim hangi noktalarda uygulanmalıdır ?
- Denetim için hangi araçlara gerek vardır ?
- Hangi tür denetime gereksinim duyulmaktadır ?
- Denetim ne kadar karmaşık olabilir ?

Kullanılan Teknikler :

a. Teknik Denetim

MARMARA ÜNİVERSİTESİ YAYINEVİ

b. Bilgisayar Denetimi
c. İstatistiksel Denetim
d. Üretim Denetimi
e. Yönetim Denetimi

B.5. Güvenilirlik

- Belirsizliğin, sistem güvenilirliği üzerindeki etkisi göz önüne alındı mı ?
- Güvensizlik kabul edilebilir bir düzeye indirgenebildi mi ?

Kullanılan Teknikler : Güvenilirlik Teknikleri

C. TASARIMI YAPILAN SİSTEMİN KURULMASI - SİSTEM KURMA

C.1. Belgeleme

- Sonuçlar üzerinde uzlaşıldı mı ?
- Rapor en olumlu etkiyi sağlayacak yeterlilik yazıldı mı?
- Sistemin kullanıcıları tam anlamıyla çalışma ile bütünleştirildi mi ?

Kullanılan Teknikler : Rapor Yazma Teknikleri

C.2. Kurma

- Sisteme ilişkin tüm somut ve soyut nesneler tamamen hazırlandı mı ?
- Sistemin kullanıcıları, sistem tasarımının felsefesi ile sürekli ilişki halinde tutuldular mı ?
- Sistemin kurulma projesi iyi planlanıp, hedefler açık olarak belirlendi mi ?

Kullanılan Teknikler : CPM- Critical Path Method

D. KURULAN YENİ SİSTEMİN İŞLETİLMESİ VE İZLENMESİ - SİSTEMİN İŞLETİLMESİ

Sistem tasarımlandıktan, kurulduktan ve yerleştirildikten sonra, sıra sistemin işlemesine gelmiştir. Sistemin işlemesi üç adımı içermektedir.

D.1. Başlangıç İşlemi

- Başlangıç için gerekli planlar yapıldı mı ?
- Sistem kullanıcıları ile etkili ve verimli bir ilişki kurulabildi mi ?
- Tüm işlemsel pürüzler düzeltildi mi ?

Kullanılan Teknikler : Sistem Analizi Teknikleri

MARMARA ÜNİVERSİTESİ YAYINEVİ

D.2. Projenin Değerlendirilmesi

Yeni sistem yeterince uzun bir süre işletildikten sonra, sistem gurubu ile sistem kullanıcıları bir araya gelerek sistem performansını değerlendirmelidir. Eğer sistem planlara uygun bir şekilde işliyor ise, başlangıçta ki sistem düşüncesi uygulama gerçekleri ile uyuşuyor demektir. Öte taraftan, eğer, sistem performansı beklenen düzeyde değil ise, o zaman yeni çabaların gerekeceğini unutmamak gerekir.

- Mevcut performans, tasarım aşamasında öngörülen performans düzeyinde midir ?
- Eğer değilse, neden ?
- Değerlendirme yeterince belgelendi mi ?

D.3. İyileştirilmiş İşlem

İyileştirilmiş işlem aşağıda ki iki durumda kendini gösterir.

- Eğer, sistemin değerlendirilmesi sonucu, mevcut performansın planda belirtilenin altına düştüğünü gösteriyor ise,
- Belli durumlarda, belli parametrelerin, yalnızca, işlemsel deneyim kazanıldıktan sonra optimize edilebilmeleri nedeniyle,

Bu durumlarda, sistemin çevresine "ayarlanması" gerekir. Eğer orijinal sistemin tasarımı yeterli doyuruculukta yapıldı ise ayarlama çalışması fazla zorlanmayı gerektirmeyecektir.

- Sistemin yeniden optimizasyona gereksinimi var mı ?
- Bu en iyi biçimde nasıl sağlanır ?
- Sonuçta elde edilen iyileştirilmiş işlem yeterli midir ?

Kullanılan Teknikler : Sistem Tasarım Teknikleri

5.2. SİSTEM YAKLAŞIMINDA KULLANILAN ARAÇLAR

Sistem yaklaşımında kullanılan araçlar şu şekilde sıralanabilir.

- Sistem Akış Diyagramları
- Program Akış Diyagramları
- Belge Akış Diyagramları
- Veri Akış Diyagramları
- Karar Tabloları

Bu araçlar, bir yöntem özelliği taşımaktan çok, sistemler arası ilişkileri en iyi gösteren ve bilgisayar destekli sistem yaklaşımında yararlanılan araçlardır.

5.2.1. SİSTEM AKIŞ DİYAGRAMI

Sistem akış diyagramları, girdinin, sisteme nereden girdiğini, nasıl işlendiğini, nasıl denetlendiğini ve bir çıktı olarak veya bir birikim biçiminde sistemi nasıl terk ettiğini gösterirler. Sistem akış diyagramı, herhangi bir işletme veya büroda elle mekanik olarak veya elektronik bir biçimde gerçekleştirilen tüm faaliyetlerin grafik temsiline verilen genel isimdir.

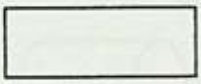
Bu nedenle bir sistem akış diyagramı;

- bilgilerin kaynağını,
- işlem sürecine aktarılış şeklini,
- işlem sürecini,
- sonuçların hangi ortam aracılığı ile dış ortama aktarılacağını ve
- yapısını gösteren çok kıymetli bir belgedir.

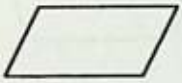
Böylece bir sistem akış diyagramı; personel, bilgi ve teçhizat arasında ki ilişkileri görsel bir biçimde gösterir. Bu tip akış diyagramlarında standart bir yapı olmamakla beraber, bu konuda kullanılan sembollerin standart olduğu söylenebilir.

5.2.1.1. SİSTEM AKIŞ DİYAGRAMI SEMBOLLERİ

Sistem akış diyagramlarında kullanılan semboller ve anlamları şu şekildedir.



İŞLEM (PROCESS) — İşlem veya işlemler kümesinin yapıldığını gösterir.



GENEL GİRİŞ/ ÇIKIŞ — Genel bilgi giriş ve çıkışını temsil eder.



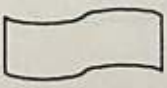
DELİKLİ KART (PUNCHED CARD) — Bilgisayara veri girişinin veya bilgisayardan ve veri çıkışının delikli kart ortamında yapıldığını gösterir.



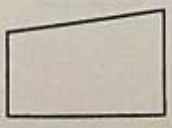
BELGE (DÖKÜMAN) (DOCUMENT) — Girdi veya çıktının basılmış bir belge veya bir rapor olduğunu gösterir.



MANYETİK BAND (MAGNETIC TAPE) — Girdi- çıktı verisinin manyetik band ortamında bulunduğunu gösterir.

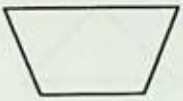


DELİKLİ ŞERİT (PUNCHED INPUT) — Girdi- çıktı verisinin delikli kâğıt veya plastik band ortamında olduğunu gösterir. Delikli kâğıt veya plastik band ile bilgisayara bilgi girişini veya yine delikli kart veya plastik band halinde bilgisayardan bilgi çıkışını temsil eder

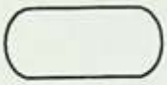


ELLE GİRİŞ (MANUAL INPUT) — Giriş verisinin, işlem sırasında bir araç kullanılarak el ile girildiğini gösterir.

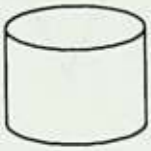
MARMARA ÜNİVERSİTESİ YAYINEVİ



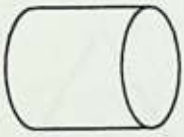
ELLE İŞLEM (MANUAL OPERATION) Mekanik bir araç kullanıma gerek kalmadan, doğrudan elle yapılan işlemleri gösterir.



ALETLE İŞLEM Kart delme vb. gibi klavye kullanılarak yapılan işlemleri gösterir.



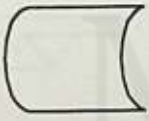
MANYETİK DİSK (MAGNETIC DISK) Girdi- çıktı verisinin manyetik disk ortamında bulunduğunu gösterir.



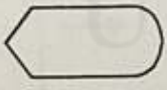
MANYETİK TAMBUR (MAGNETIC DRUM) Girdi- çıktı verisinin manyetik tambur ortamında bulunduğunu gösterir.



BAĞSIZ BELLEK (OFFLINE STORAGE) Bilgisayarla doğrudan ilişkilendirilmemiş (bilgisayarın doğrudan ulaşamadığı) kâğıt şerit, delikli kart, manyetik band, kâğıt belgeler vb. bellek ortamlarını temsil eder.



BAĞLI BELLEK (ONLINE STORAGE) Bilgisayar tarafından doğrudan ulaşılan manyetik disk, tambur ve benzeri bellek birimlerini temsil eder.

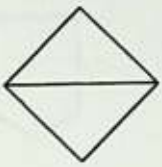


GÖRÜNTÜLEME (DISPLAY) İşlem sırasında veya sonunda bilgilerin ekranda görüntülendiğini belirtir.



BİRLEŞTİRME VE ÇIKARMA İki veya daha fazla bilgi kümesinden iki veya daha fazla yeni bilgi kümesinin birleştirme veya çıkarma yoluyla oluşturulduğunu gösterir.

MARMARA ÜNİVERSİTESİ YAYINEVİ



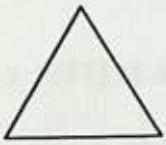
SIRALAMA (SORT)

Sıralama işleminin yapıldığını gösterir.



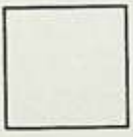
BİRLEŞTİRME (MERGE)

İki veya daha fazla bilgi kümesinin tek bir bilgi kümesi halinde toplandığını gösterir.
Bu sembol iki ayrı veri dosyasının içindeki bilgilerin düzeni bozulmayacak şekilde, birleştirilip tek bir dosya haline getirilmesi işlemini temsil eder.



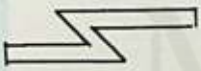
ÇIKARMA (EXTRACT)

Bir bilgi kümesinden, bir veya daha fazla bilgi kümesi elde edildiğini (çıkarıldığını) gösterir.



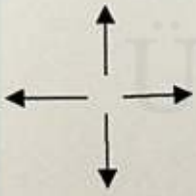
YARDIMCI İŞLEM (AUXILARY OPERATION)

Bu sembol merkezi işlem biriminin doğrudan denetimi altında olmayan bağsız birimlerin yaptığı işlemi tarif eder.



İLETİŞİM HATTI (COMMUNICATION LINE)

Bilginin, iletişim hatları aracılığı ile bir başka yere o yerden bilgi Alındığını gösterir.



AKIŞ (ARROWHEADS)

İşlem veya veri akışının yönünü gösterir.



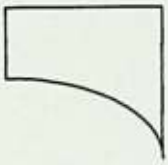
SAYFA İÇİ BAĞLANTI (CONNECTER) (ON- PAGE CONNECTOR)

Bir sayfada ki bir noktanın, aynı sayfada ki bir diğer noktaya bağlantısını gösterir

MARMARA ÜNİVERSİTESİ YAYINEVİ



SAYFALAR ARASI BAĞLANTI (OFF-PAGE CONNECTOR) Bir sayfadaki bir noktanın, bir diğer sayfada ki bir noktaya bağlantısını gösterir.



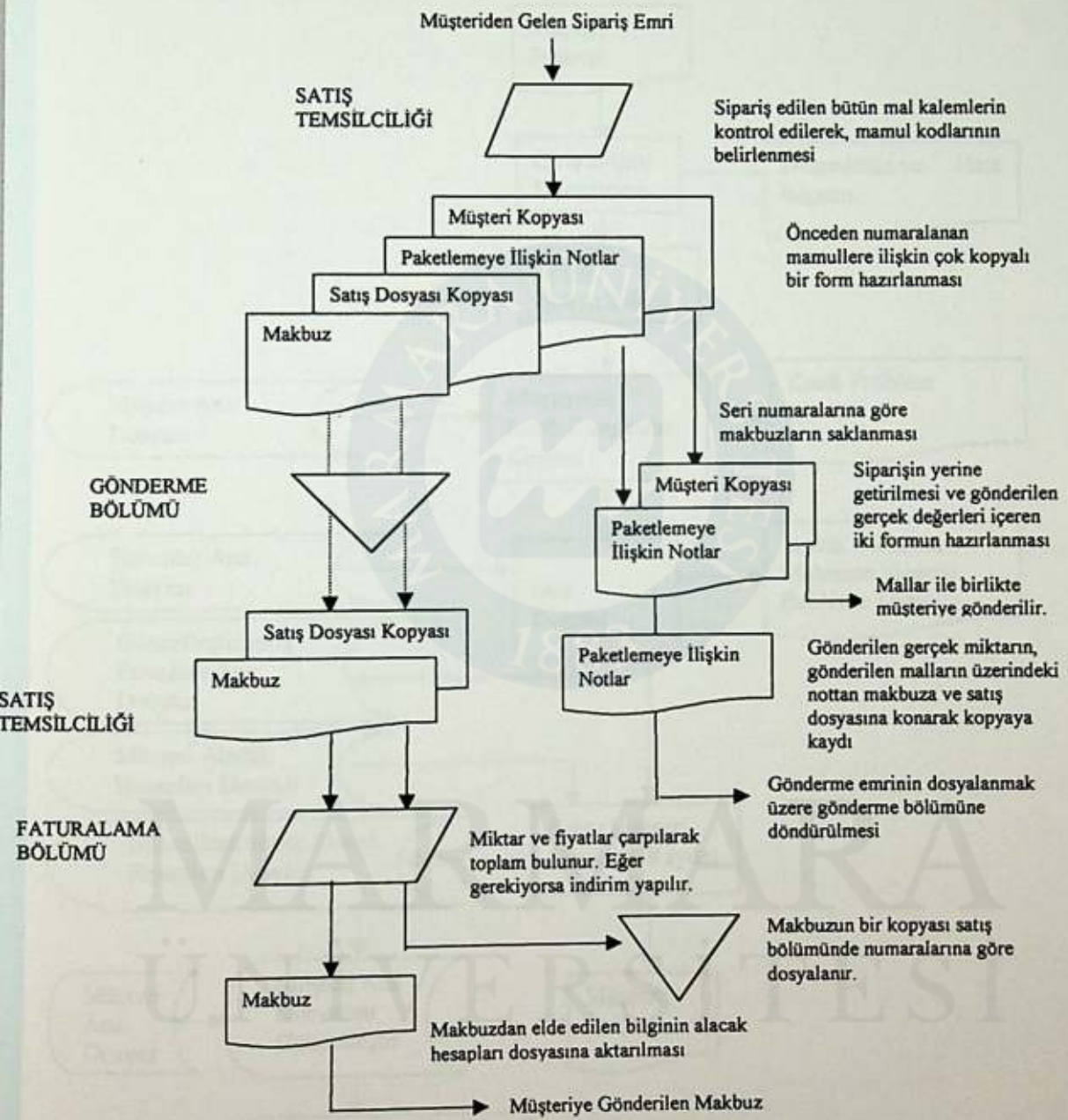
GEÇİŞ ŞERİDİ (TRANSMITTAL TAPE) Bilgilerin küme halinde denetiminin ve irdelenmesinin yapılmasını sağlayan geçiş bandının kullanıldığını gösterir.

Şekil 8 ve 9'da "Bir Sipariş Emrinin İşlemesine İlişkin Sistem Akış Diyagramı" örnekleri verilmiştir.

5.2.1.2. SİSTEM AKIŞ DİYAGRAMI ÖRNEKLERİ

Sistem akış diyagramlarına ilgili olarak 2 sistem akış diyagramı örneği aşağıda verilmiştir.

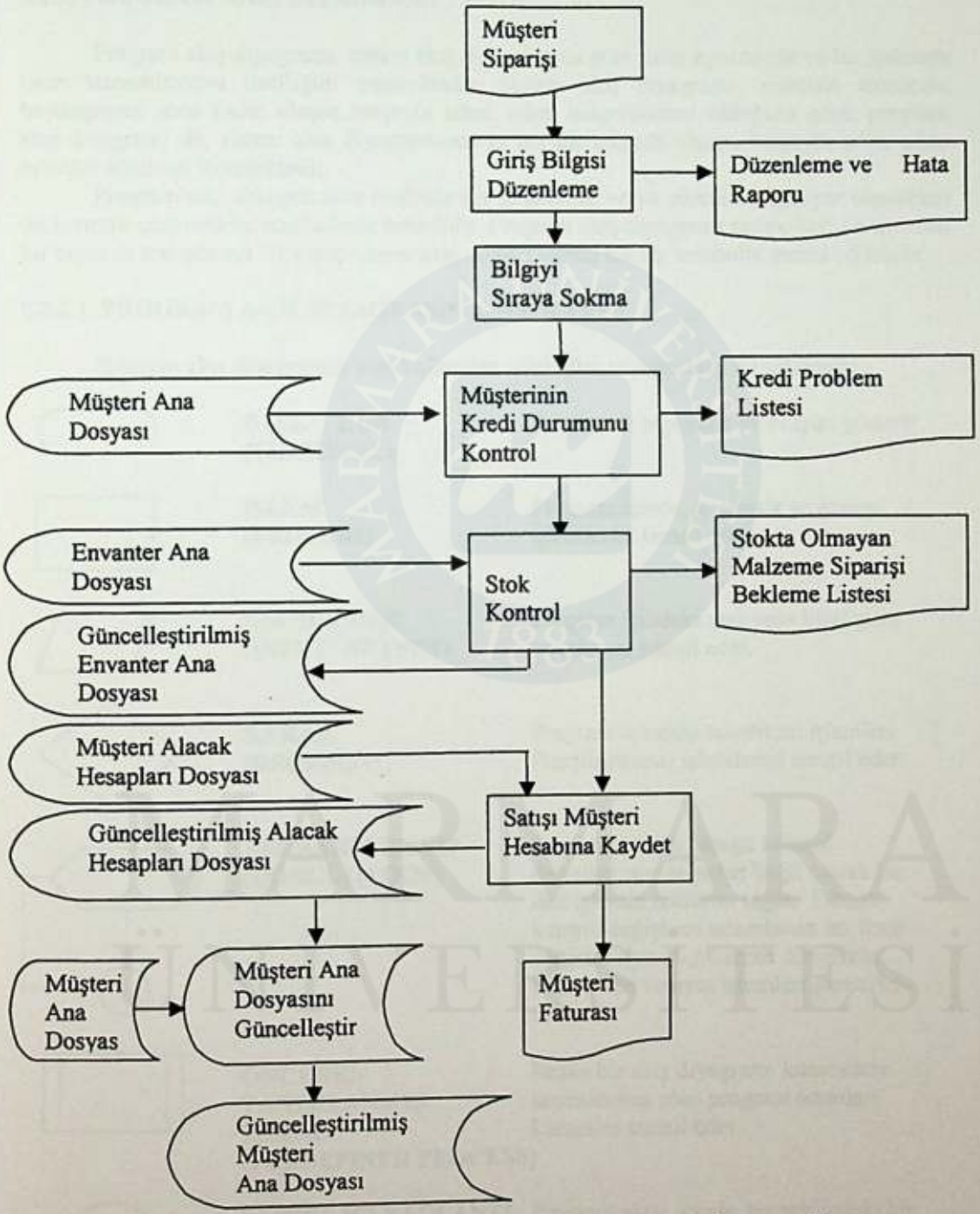
MARMARA ÜNİVERSİTESİ YAYINEVİ



Şekil 8. Bir Sipariş Emrinin İşlemesine İlişkin Sistem Akış Diyagramı Örneği I[13]

[13] Esen, Öner, "Programlamaya Giriş ve Basic", s.38.

MARMARA ÜNİVERSİTESİ YAYINEVİ



Şekil 9. Bir Sipariş Emrinin İşlemesine İlişkin Sistem Akış Diyagramı Örneği II [14]

[14] Ertürk, Zeki, "Sistem Analiz" , Ekonomist yayınevi, Ankara, 1983, s.47.

5.2.2. PROGRAM AKIŞ DİYAGRAMI

Program akış diyagramı, sistem akış diyagramına göre daha ayrıntılıdır ve bir anlamda onun tamamlayıcısı özelliğini taşımaktadır. Sistem akış diyagramı, sistemin tümünde, başlangıçtan sona kadar oluşan herşeyin adım, adım belgelenmesi olduğuna göre, program akış diyagramı da, sistem akış diyagramında ki her bir adımda oluşan herşeyin adım adım ayrıntılı analizini içermektedir.

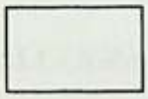
Program akış diyagramları özellikle bir problemin ve ya sürecin bilgisayar olanakları ile ayrıntılı çözümünün sembollerle temsilidir. Program akış diyagramı sembolleri ile ayrıntılı bir biçimde tanımlanan bir süreç sistem akış diyagramında tek bir sembolle temsil edilebilir.

5.2.2.1. PROGRAM AKIŞ DİYAGRAMI SEMBOLLERİ

Program akış diyagramlarında kullanılan semboller ve anlamları şu şekildedir.



BAŞLA/ DUR (TERMİNAL) — Programın başlangıç ve bitişini gösterir



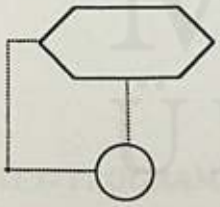
İŞLEM (PROCESS) — Program içinde, aritmetik ve atama işlemlerini temsil eder.



GİRDİ/ ÇIKTI (INPUT/ OUTPUT) — Program içindeki veri veya bilgi giriş ve çıkışını temsil eder.



KARAR (DECISION) — Program içindeki mantıksal işlemleri (karşılaştırma) işlemlerini temsil eder.



TEKRARLI İŞLEM (DÖNGÜ) (LOOP) — Program içinde, döngü kontrol değişkeninin değerine bağlı olarak bir dizi işlemin tekrarını sağlar. Döngü kontrol değişkeni tanımlanan üst limit değerini aşmadığı sürece döngünün başına dön ve aynı işlemleri gerçekleştir.



ÖNCEDEN TANIMLANMIŞ İŞLEM (PREDEFINED PROCESS) — Başka bir akış diyagramı kümesinde tanımlanmış plan program adımları kümesini temsil eder.



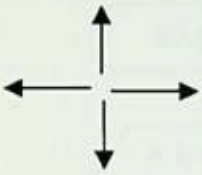
SAHİFE İÇİ BAĞLANTI (ON-PAGE CONNECTOR) — Program akışı içinde, bir sahifedeki bir noktanın, aynı sahifedeki bir diğer noktaya bağlantısını gösterir.

MARMARA ÜNİVERSİTESİ YAYINEVİ



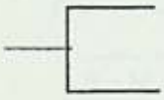
SAHİFELER ARASI BAĞLANTI
(OFF-PAGE CONNECTOR)

Program akışı içinde, bir sahifedeki bir noktanın, bir diğer sahifedeki bir noktaya bağlantısını gösterir.



AKIŞ
(ARROWEHEADS)

Program içindeki işlem akışının yönünü gösterir.



AÇIKLAMA
(COMMENT)
(ANNOTATION)

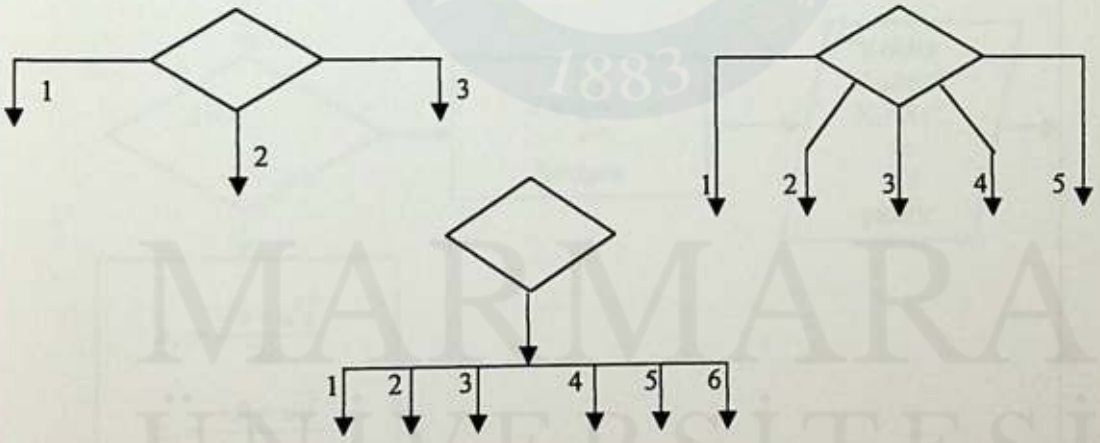
Tanımlayıcı yorumların ve açıklayıcı bilgilerin verilmesini sağlar.



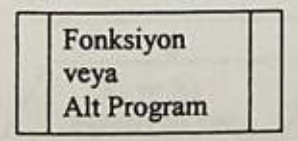
YAZICIDAN ÇIKIŞ
(PRINTER OUTPUT)

Yazıcıdan çıkış alma

DALLANMA :



ALT- PROGRAM :



5.2.2.2. PROGRAM AKIŞ DİYAGRAMI VE C PROGRAMLAMA DİLİ ÖRNEKLERİ

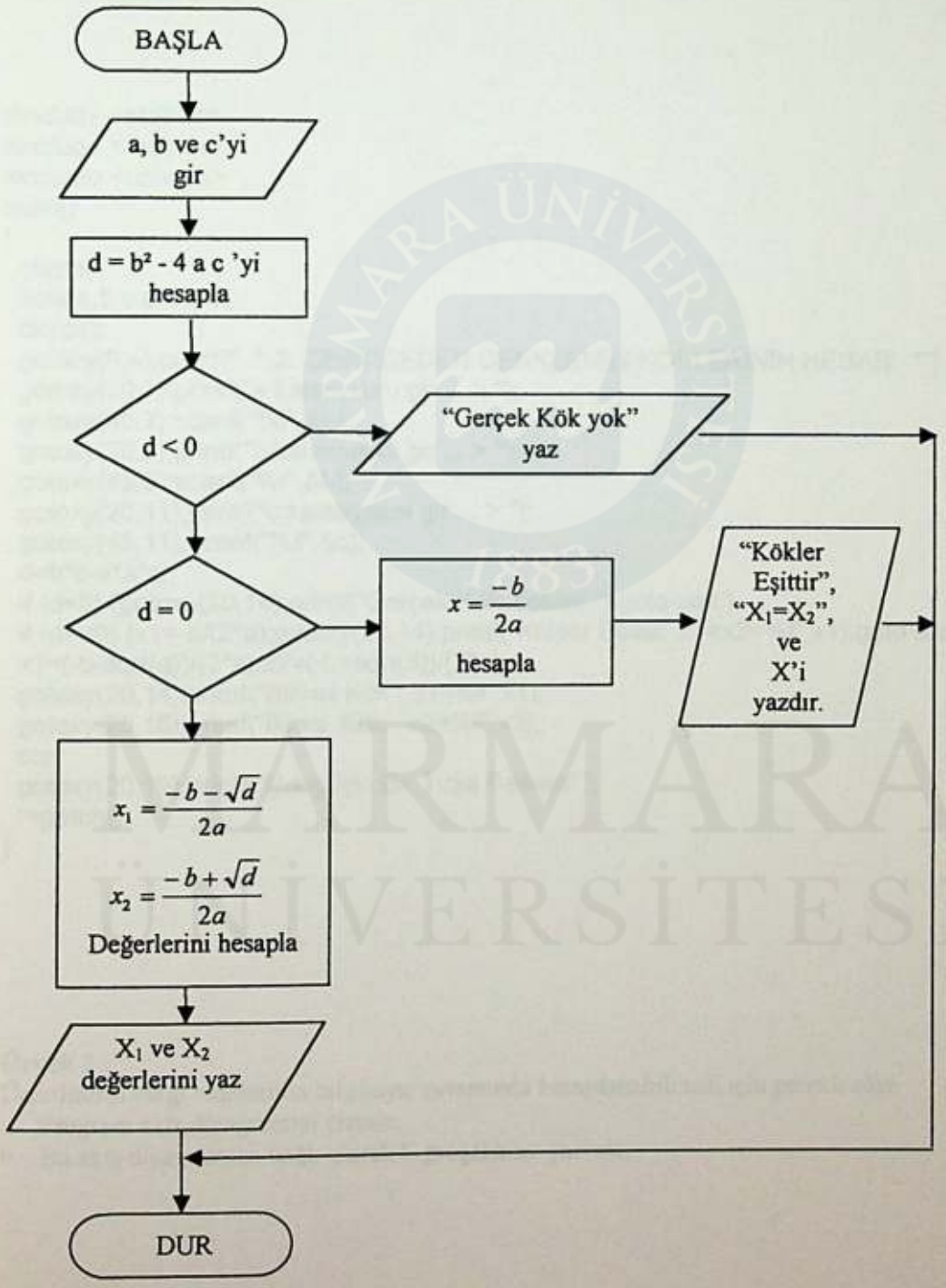
Bu bölümde çeşitli örnek olaylar ele alınmıştır. İlk aşamada, bu olayların bilgisayar ortamında çözümüne esas olacak akış diyagramı çizilmiştir. İkinci aşamada ise çizilen akış diyagramına bağlı olarak bilgisayar programı, C programlama dili kullanarak yazılmıştır.

Örnek 1 :

İkinci dereceden bir denklemin bilgisayar ortamında çözümü için gerekli olan

a. Program akış diyagramını çiziniz.

b. Bu akış diyagramına bağlı olarak C programını yazınız.



MARMARA ÜNİVERSİTESİ YAYINEVİ

b) İkinci dereceden bir denklemin köklerinin hesaplanmasını gerçekleştiren C Programı :

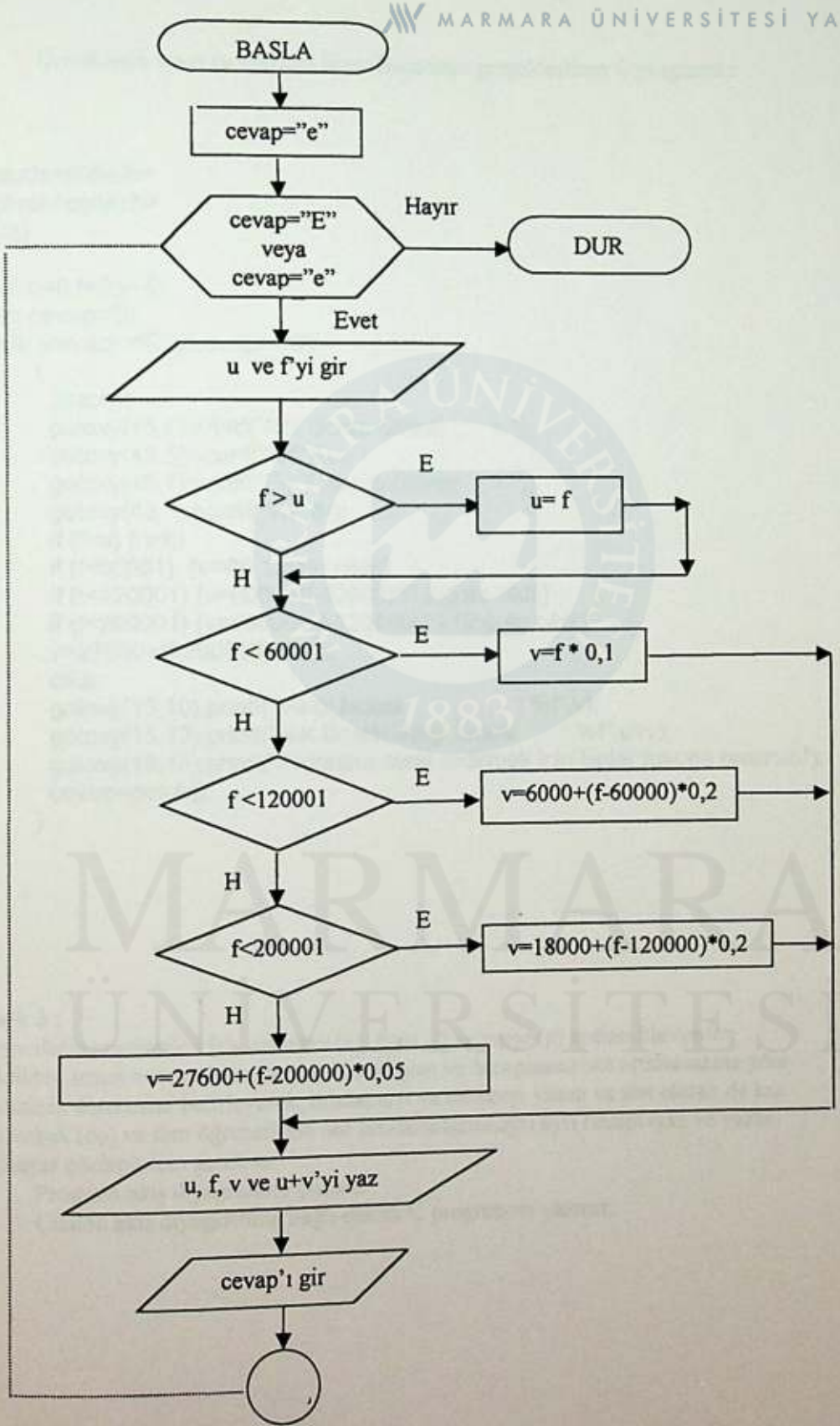
```
#include <stdio.h>
#include <math.h>
#include <conio.h>
main()
{
  char t;
  float a,b,c,d,x1,x2;
  clrscr();
  gotoxy(9,4);printf("  *  2. DERECEDEN DENKLEMİN KÖKLERİNİN HESABI  *");
  gotoxy(20,7);printf("a katsayısını gir ... > ");
  gotoxy(45,7);scanf("%f",&a);
  gotoxy(20,9);printf("b katsayısını gir ... > ");
  gotoxy(45,9);scanf("%f",&b);
  gotoxy(20,11);printf("c katsayısını gir ... > ");
  gotoxy(45,11);scanf("%f",&c);
  d=b*b-4*a*c;
  if (d<0) {gotoxy(20,14);printf("Gerçek Kök Yoktur.");goto son;}
  if (d==0) {x1=-b/(2*a);gotoxy(20,14);printf("Kökler Eşittir. x1=x2=%f",x1);goto son;}
  x1=(-b-sqrt(d))/(2*a);x2=(-b+sqrt(d))/(2*a);
  gotoxy(20,14);printf("Birinci Kök : x1=%f",x1);
  gotoxy(20,16);printf("İkinci  Kök : x2=%f",x2);
  son:
  gotoxy(20,19);printf("Çıkış İçin Bir Tuşa Basınız");
  t=getch();
}
```

Örnek 2 :

Ücretlilerin vergi iadelerinin bilgisayar ortamında hesaplanabilmesi için gerekli olan

a. Program akış diyagramını çiziniz.
b. Bu akış diyagramına bağlı olarak C programını yazınız.

MARMARA ÜNİVERSİTESİ YAYINEVİ



b) Ücretlilerin vergi iadelerinin hesaplanmasını gerçekleştiren C programı :

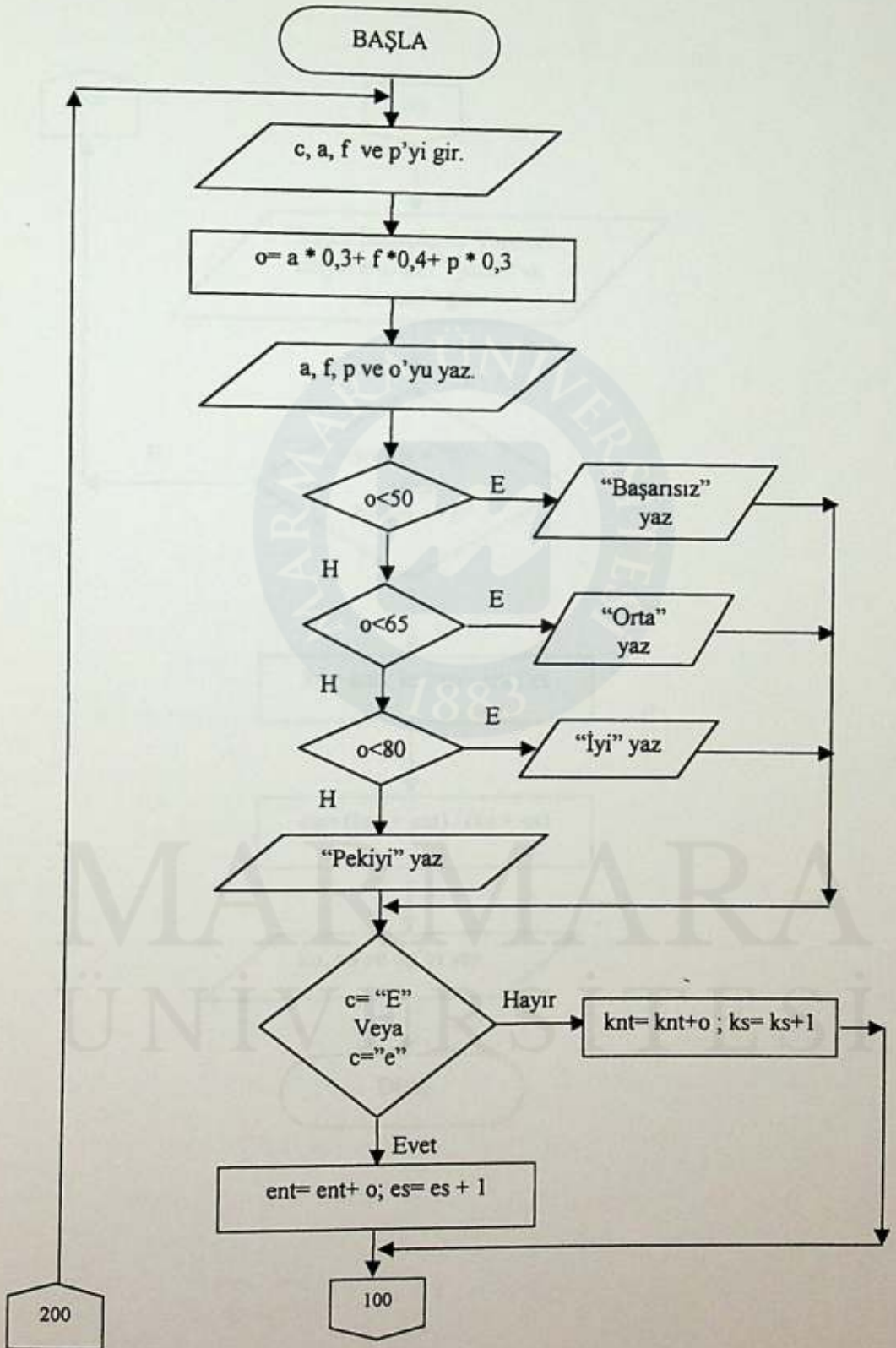
```
#include<stdio.h>
#include<conio.h>
main()
{
  float u=0,f=0,v=0;
  char cevap='E';
  while (cevap=='E' || cevap=='e')
        {
          clrscr();
          gotoxy(15,5);printf("Net Ücreti Giriniz ..... > ");
          gotoxy(43,5);scanf("%f",&u);
          gotoxy(15,7);printf("Fiş Tutarını Giriniz ... > ");
          gotoxy(43,7);scanf("%f",&f);
          if (f>u) {u=f;}
          if (f<60001)  {v=f*0.1;goto cikti;}
          if (f<120001) {v=6000+(f-60000)*0.2;goto cikti;}
          if (f<200001) {v=18000+(f-120000)*0.12;goto cikti;}
          v=27600+(f-200000)*0.05;
          cikti:
          gotoxy(15,10);printf("Vergi İadesi              : %f",v);
          gotoxy(15,12);printf("Net Ücret+Vergi İadesi     : %f",u+v);
          gotoxy(15,15);printf("Programı sona erdirmek için Enter tuşuna basınız.");
          cevap=getch();
        }
}
```

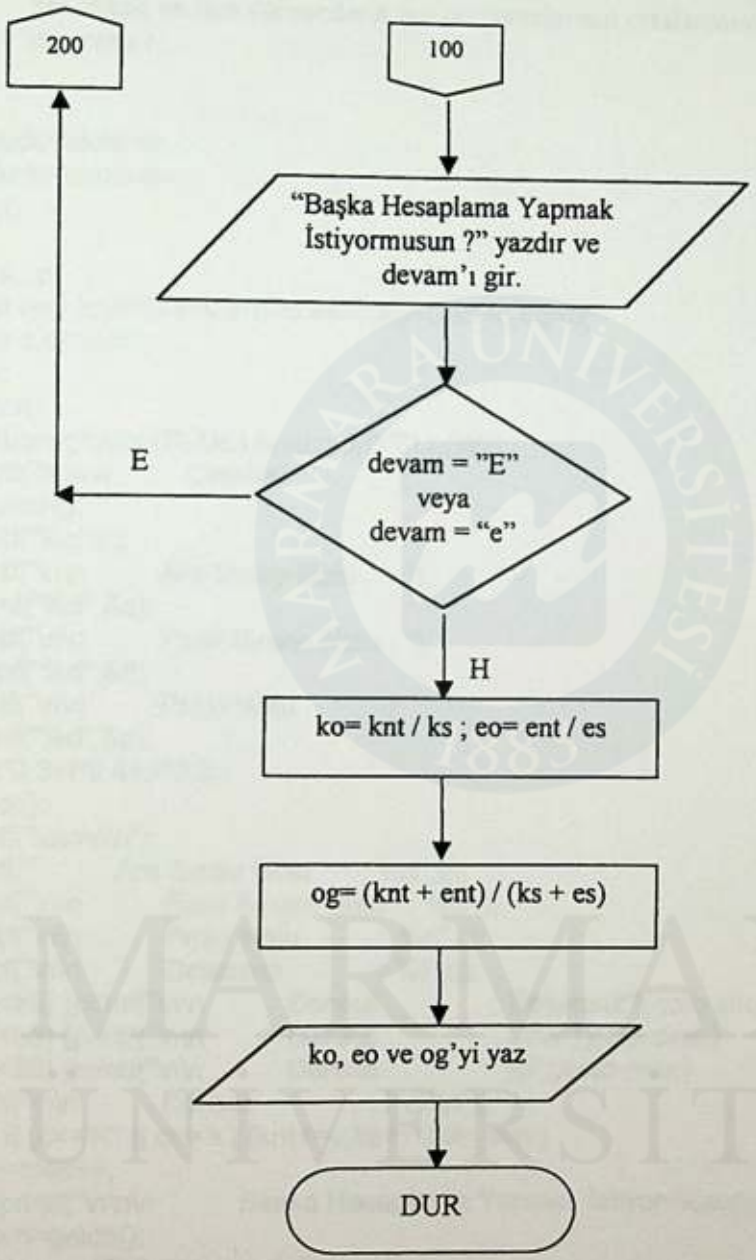
Örnek 3 :

Öğrencilerin cinsiyetleri (c), arasınav (a), final (f) ve proje (p) notları klavyeden girildikten sonra not ortalamasını (o) hesaplayan ve hesaplanan not ortalamasına göre öğrencinin derecesini belirleyerek, ortalamayı ve dereceyi yazan ve son olarak da kız (ko), erkek (eo) ve tüm öğrencilerin not ortalamalarını ayrı ayrı hesaplayan ve yazan bilgisayar çözümü için gereken

a. Program akış diyagramını çiziniz.
b. Çizilen akış diyagramına bağlı olarak C programını yazınız.

MARMARA ÜNİVERSİTESİ YAYINEVİ




200
100
"Başka Hesaplama Yapmak İstiyormusun ?" yazdır ve devam'ı gir.
E
devam = "E" veya devam = "e"
H
ko= knt / ks ; eo= ent / es
og= (knt + ent) / (ks + es)
ko, eo ve og'yi yaz
DUR

MARMARA ÜNİVERSİTESİ YAYINEVİ

b) Öğrencilerin not ortalamalarını ve derecelerini ayrı ayrı hesapladıktan sonra, kız, erkek ve tüm öğrencilerin not ortalamalarının ortalamasını hesaplayan C programı :

```
#include<stdio.h>
#include<conio.h>
main()
{
 int a,f,p;
 float o=0,knt=0,ks=0,ent=0,es=0,ko=0,eo=0,og=0;
 char c,devam;
 bas:
 clrscr();
 textbackground(BLUE);textcolor(YELLOW);
 printf("\n\n\n        Cinsiyet        : ");
 c=getch();
 printf("%c",c);
 printf("\n\n        Ara Sınav Notu    : ");
 scanf("%d",&a);
 printf("\n\n        Final Sınavı Notu : ");
 scanf("%d",&f);
 printf("\n\n        Proje Notu        : ");
 scanf("%d",&p);
 o=a*0.3+f*0.4+p*0.3;
 clrscr();
 printf("\a\n\n\n\n");
 printf("        Ara Sınav Notu     : %d",a);
 printf("\n\n        Final Sınavı Notu  : %d",f);
 printf("\n\n        Proje Notu         : %d",p);
 printf("\n\n        Ortalama           : %f",o);
 if (o<50) {printf("\n\n        Derece             : Basarısız");goto cins;}
 if (o<65) {printf("\n\n        Derece             : Orta");goto cins;}
 if (o<80) {printf("\n\n        Derece             : İyi");goto cins;}
 printf("\n\n        Derece          : Pekiyi");
 cins:if (c=='K'  || c=='k') {knt+=o;ks++;goto son;}
 ent+=o;es++;
 son:printf("\n\n\n        Baska Hesaplama Yapmak İstiyormusunuz ? (E/H) ");
 devam=getch();
 if (devam=='E' || devam=='e') goto bas;
 clrscr();
 ko=knt/ks;eo=ent/es;og=(knt+ent)/(ks+es);
 printf("\n\n\n        Kızların Not Ortalaması    : %f",ko);
 printf("\n\n        Erkeklerin Not Ortalaması  : %f",eo);
 printf("\n\n\n        Genel Ortalama           : %f",og);
 printf("\n\n\n        Bir tuşa basınız.....>");
 devam=getch();
}
```

MARMARA ÜNİVERSİTESİ YAYINEVİ

Örnek 4 :

2'den 20'ye kadar 3'er atlayarak bulunacak sayıları,

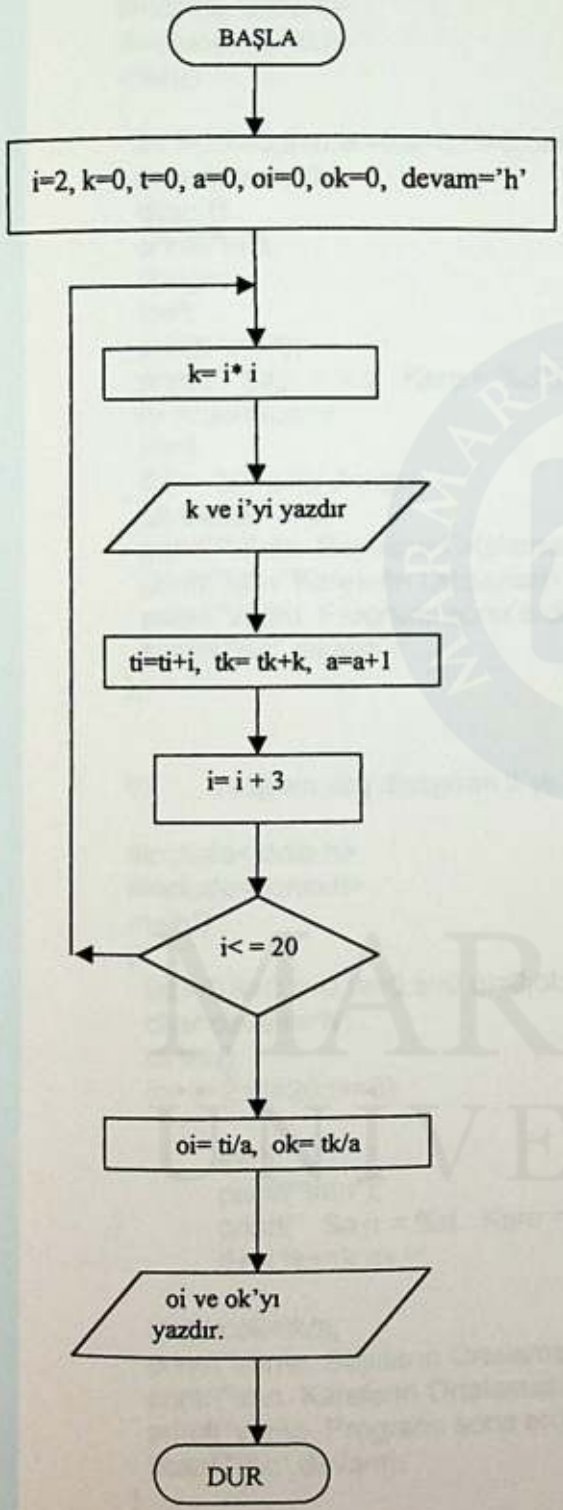
- Üretecek ve ekranda yazacak,
- Üretilen sayıların karelerini hesaplayacak ve ekranda yazacak,
- Sayıların ve karelerin ortalamalarını hesaplayarak, ekranda yazacak bilgisayar programı için,

a. Program akış diyagramlarını, karar ve döngü sembollerini kullanarak ayrı ayrı çiziniz.

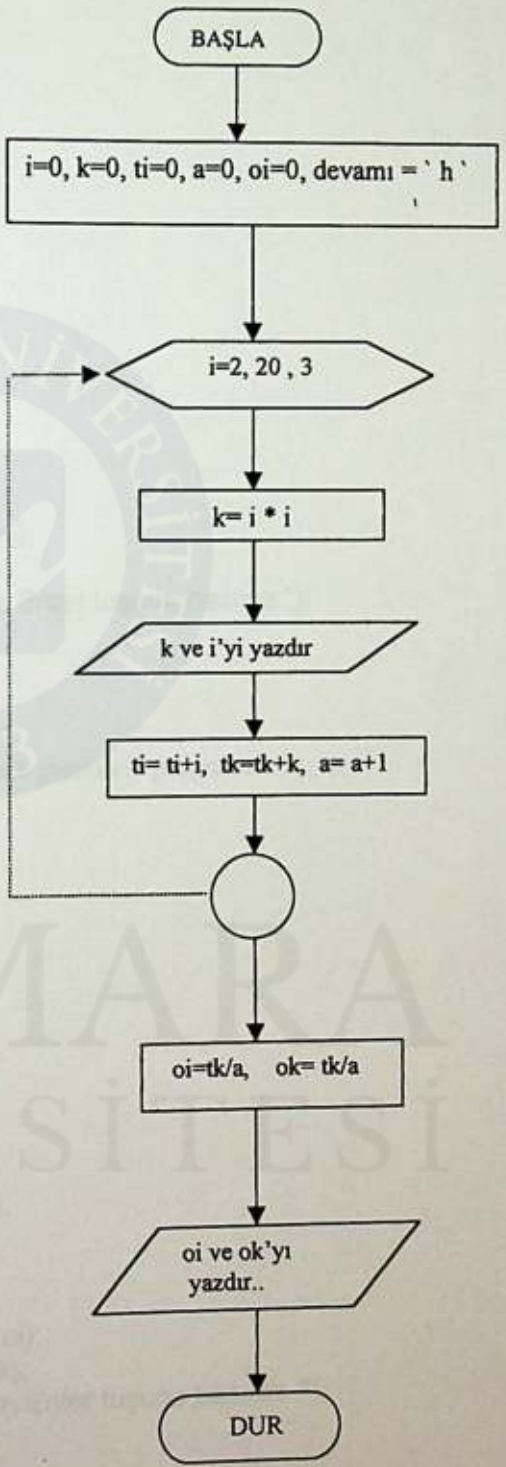
b. Çizilen akış diyagramlarına bağlı olarak ve C programlama dilini kullanarak gerekli olan bilgisayar programlarını yazınız.

MARMARA ÜNİVERSİTESİ YAYINEVİ

Akış Diyagramı 1 :



Akış Diyagramı 2 :



MARMARA ÜNİVERSİTESİ YAYINEVİ

b) Program akış diyagramı 1'e bağlı olarak yazılan C programı :

```
#include<stdio.h>
#include<conio.h>
main()
{
 int i=2,k=0,ti=0,tk=0,a=0,oi=0,ok=0;
 char devam='h';
 clrscr();
 printf("\n");
 dongu:
 k=i*i;
 printf("\n\n");
 printf("  Sayı = %d   Kare = %d",i,k);
 ti+=i;tk+=k;a++;
 i+=3;
 if (i<=20) goto dongu;
 oi=ti/a;ok=tk/a;
 printf("\n\n\n  Sayıların Ortalaması : %d",oi);
 printf("\n\n  Karelerin Ortalaması : %d",ok);
 printf("\n\n\n  Programı sona erdirmek için Enter tuşuna basınız.");
 scanf("%c",devam);
}
```

b) Program akış diyagramı 2'ye bağlı olarak yazılan C programı :

```
#include<stdio.h>
#include<conio.h>
main()
{
 int i=0,k=0,ti=0,tk=0,a=0,oi=0,ok=0;
 char devam='h';
 clrscr();
 for (i=2;i<=20;i+=3)
    {
       k=i*i;
       printf("\n\n");
       printf("  Sayı = %d   Kare = %d",i,k);
       ti+=i;tk+=k;a++;
    }
 oi=ti/a;ok=tk/a;
 printf("\n\n\n  Sayıların Ortalaması : %d",oi);
 printf("\n\n  Karelerin Ortalaması : %d",ok);
 printf("\n\n\n  Programı sona erdirmek için Enter tuşuna basınız. ");
 scanf("%c",devam);
}
```

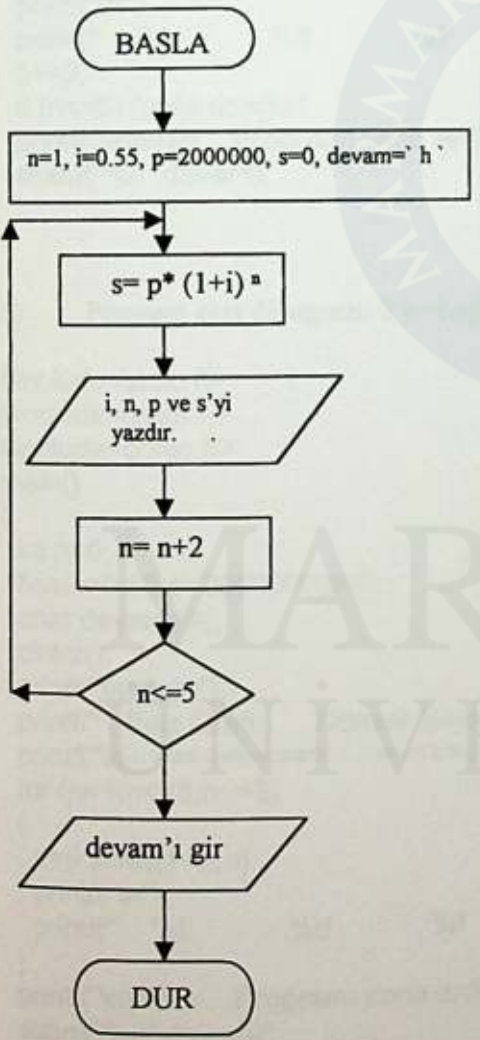
MARMARA ÜNİVERSİTESİ YAYINEVİ

Örnek 5 :

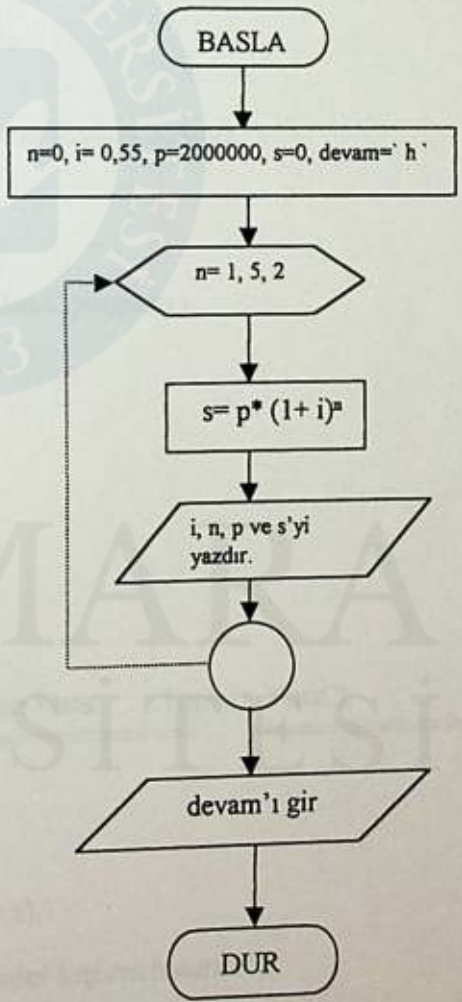
Yıllık %55 faiz oranı itibarıyle, 1, 3 ve 5 yıl süreyle 2,000,000.- TL, bankaya yatırıldığı takdirde, dönem sonunda elde edilecek toplam para miktarlarını hesaplayarak, ekranda yazacak bilgisayar programı için,

a. Gerekli olan akış diyagramlarını, karar ve döngü sembollerini kullanarak ayrı ayrı çiziniz.
b. Çizilen akış diyagramlarına bağlı olarak ve C programlama dilini kullanarak gerekli olan bilgisayar programlarını yazınız.

Akış Diyagramı 1 :



Akış Diyagramı 2 :



b) Program akış diyagramı 1'e bağlı olarak yazılan C programı :

MARMARA ÜNİVERSİTESİ YAYINEVİ

```
#include<stdio.h>
#include<math.h>
#include<conio.h>
main()
{
 int n=1;
 float i=0.55,p=2000000,s=0;
 char devam='h';
 clrscr();
 printf("\n\n\n\n");
 printf("  Faiz Oranı     Dönem Sayısı     Yatırılan Para      Toplam  Para");
 printf("\n  ==========     ==========     =============     =============");
 dongu:
 s=p*pow((1+i),n);
 printf("\n");
 printf("   %f          %d          %f      %f",i,n,p,s);
 n+=2;
 if (n<=5) {goto dongu;}
 printf("\n\n\n\n    Programı sona erdirmek için Enter tuşuna basınız.");
 scanf("%c",devam);
 }
```

b) Program akış diyagramı 2'ye bağlı olarak yazılan C programı :

```
#include<stdio.h>
#include<math.h>
#include<conio.h>
main()
{
 int n=0;
 float i=0.55,p=2000000,s=0;
 char devam='h';
 clrscr();
 printf("\n\n\n\n");
 printf("  Faiz Oranı     Dönem Sayısı     Yatırılan Para      Toplam  Para");
 printf("\n  ==========     ==========     =============     =============");
 for (n=1;n<=5;n+=2)
 {
  s=p*pow((1+i),n);
  printf("\n");
  printf("   %f          %d          %f      %f",i,n,p,s);
 }
 printf("\n\n\n\n    Programı sona erdirmek için Enter tuşuna basınız.");
 scanf("%c",devam);
}
```

MARMARA ÜNİVERSİTESİ YAYINEVİ

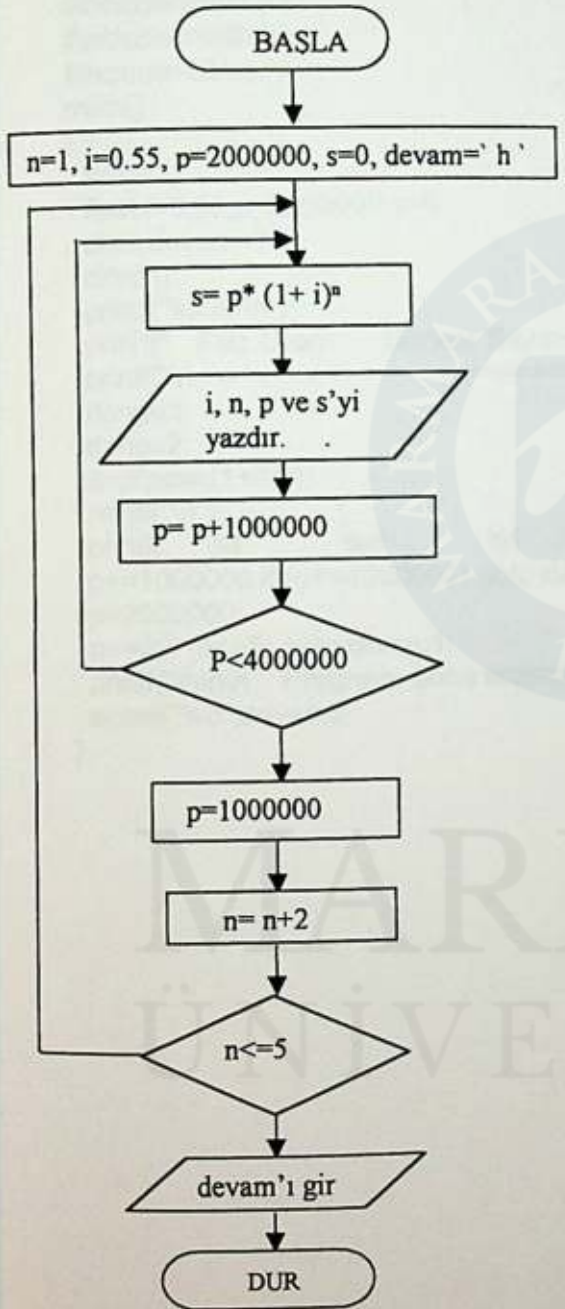
Örnek 6 :

Yıllık %55 faiz oranı itibariyle, 1, 3, ve 5 yıl süreyle 2,000,000.–TL, 3,000,000.–TL ve 4,000,000.–TL, bankaya yatırıldığı takdirde, dönem sonunda elde edilecek toplam para miktarlarını hesaplayacak bilgisayar programı için,

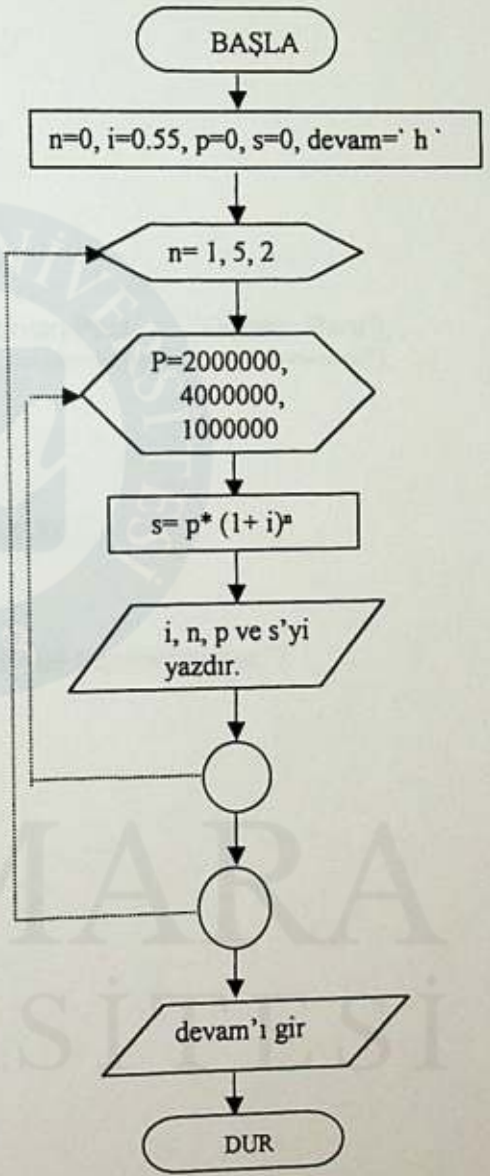
a. Gerekli olan akış diyagramlarını karar ve döngü sembollerini kullanarak ayrı ayrı çiziniz.
b. Çizilen akış diyagramlarına bağlı olarak ve C programlama dilini kullanarak gerekli olan bilgisayar programını yazınız.

MARMARA ÜNİVERSİTESİ YAYINEVİ

Akış Diyagramı 1 :



Akış Diyagramı 2 :



b) Program akış diyagramı 1'e bağlı olarak yazılan C programı :

```
#include<stdio.h>
#include<math.h>
#include<conio.h>
main()
{
 int n=1;
 float i=0.55,p=2000000,s=0;
 char devam='h';
 clrscr();
 printf("\n\n\n\n");
 printf("  Faiz Oranı     Dönem Sayısı     Yatırılan Para      Toplam  Para");
 printf("\n  ==========     ==========     ==========     =========");
 dongu1:
 dongu2:
 s=p*pow((1+i),n);
 printf("\n");
 printf("   %f          %d          %f      %f",i,n,p,s);
 p+=1000000;if (p<=4000000) goto dongu2;
 p=2000000;
 n+=2;if (n<=5) goto dongu1;
 printf("\n\n\n   Programı sona erdirmek için Enter tuşuna basınız.");
 scanf("%c",devam);
}
```

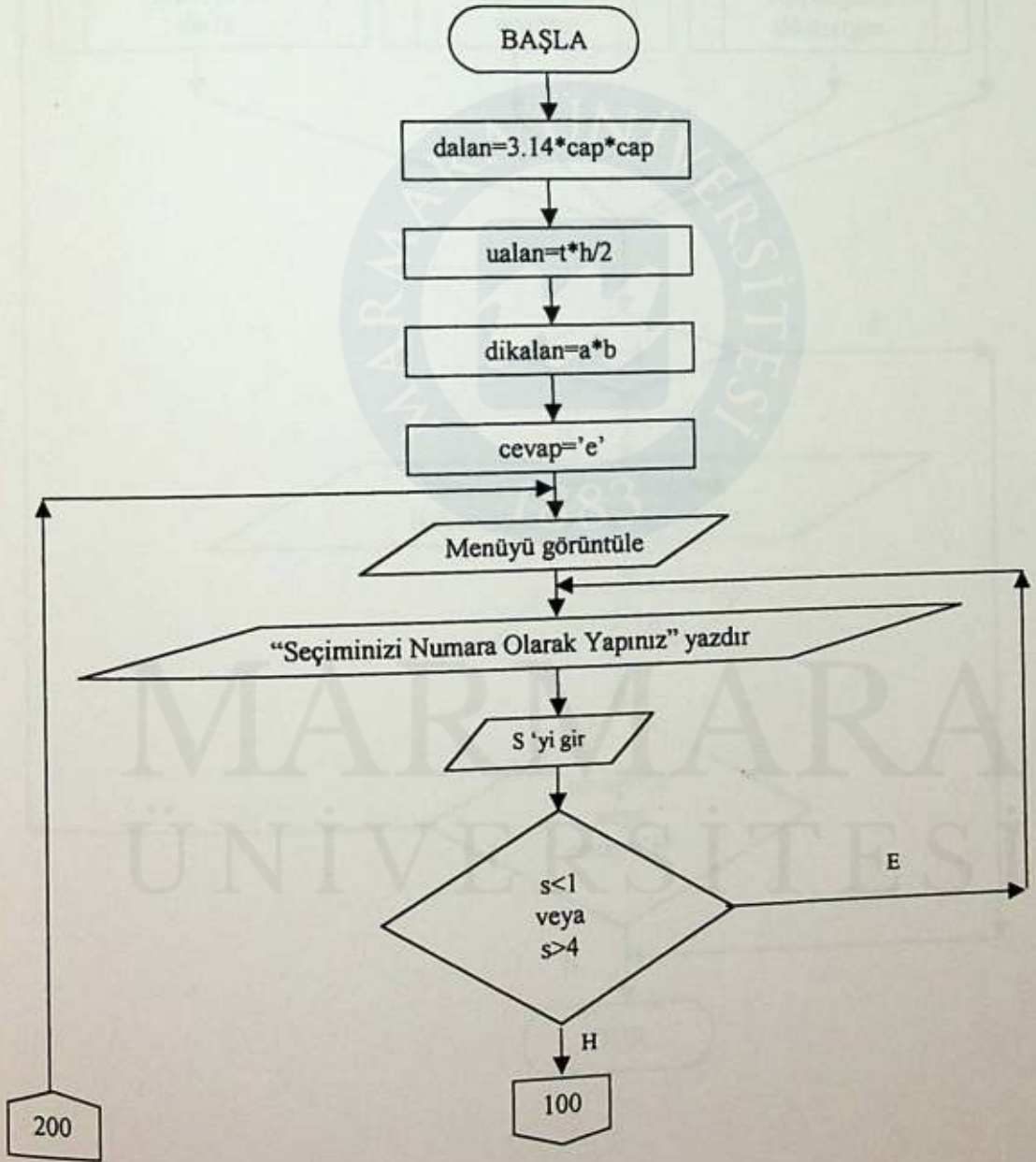
b) Program akış diyagramı 2'ye bağlı olarak yazılan C programı :

```
#include<stdio.h>
#include<math.h>
#include<conio.h>
main()
{
 int n=1;
 float i=0.55,p=2000000,s=0;
 char devam='h';
 clrscr();
 printf("\n\n\n\n");
 printf("  Faiz Oranı      Dönem Sayısı      Yatırılan Para       Toplam  Para");
 printf("\n  ==========     ==========     ===========     ========");
 for (n=1;n<=5;n+=2)
    {
       for (p=200000;p<=400000;p+=100000)
          {
           s=p*pow((1+i),n);
           printf("\n");
           printf("   %f          %d          %f       %f",i,n,p,s);
          }
    }
 printf("\n\n\n    Programı sona erdirmek için Enter tuşuna basınız.");
 scanf("%c",devam);
}
```

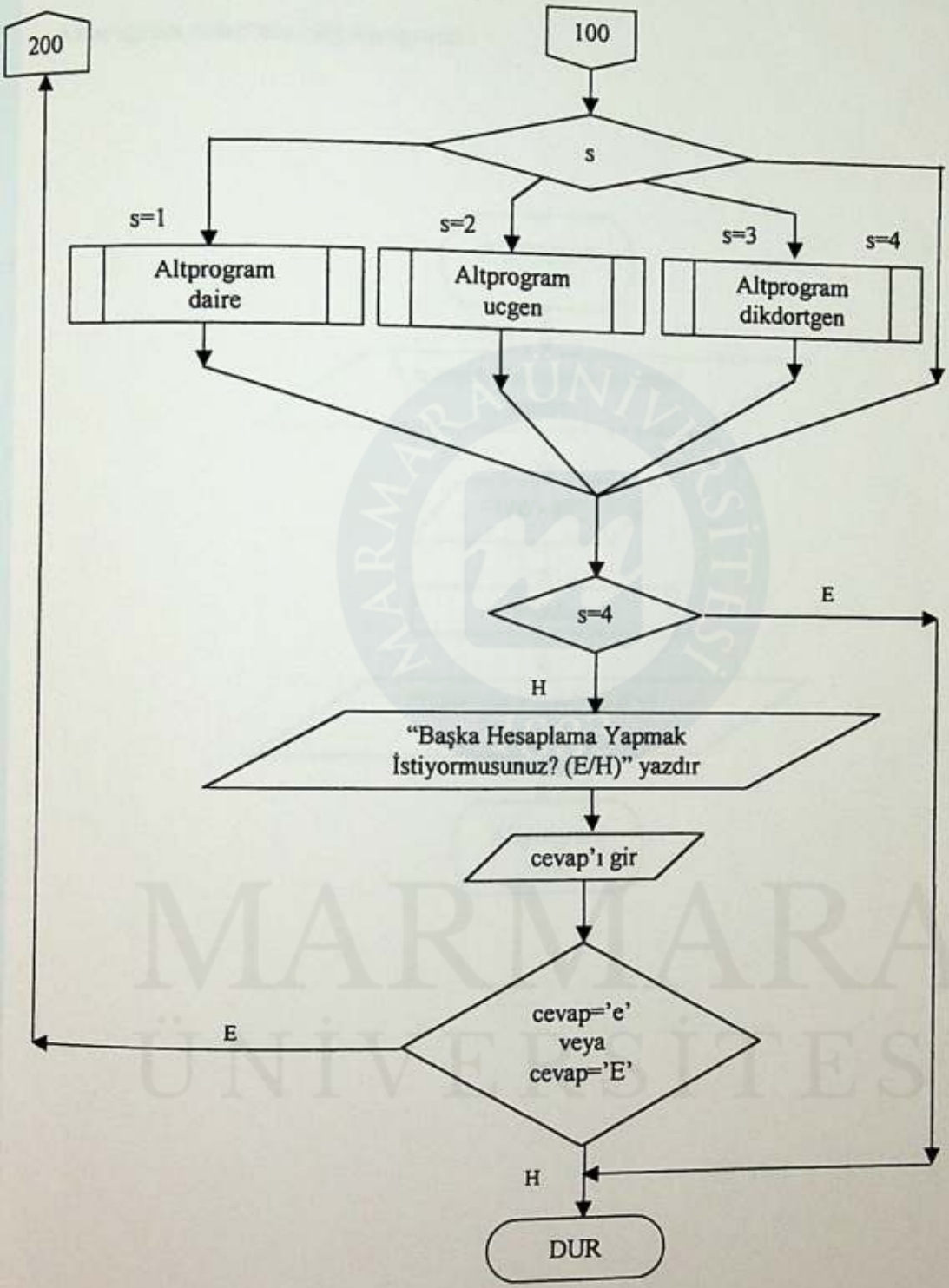
Örnek 7 :

Ekranda görüntülenecek "Alan Hesabı Menüsü" nden kullanıcının yapacağı seçime bağlı olarak, ilgili alan hesapları alt programlarına erişilmesini sağlayan ve bu alt programlarda girilen değerler itibariyle ilgili geometrik şekilin alan hesabını gerçekleştirerek ekrana yazan ve son olarak da menünün son seçeneği ile kullanıcıya programdan çıkış olanağı sağlayan bilgisayar programı için

a. Akış diyagramını çiziniz.
b. Çizilen akış diyagramına bağlı olarak ve C programlama dilini kullanarak gerekli olan bilgisayar programını yazınız.



MARMARA ÜNİVERSİTESİ YAYINEVİ



MARMARA ÜNİVERSİTESİ YAYINEVİ

Altprogram daire'nin akış diyagramı :



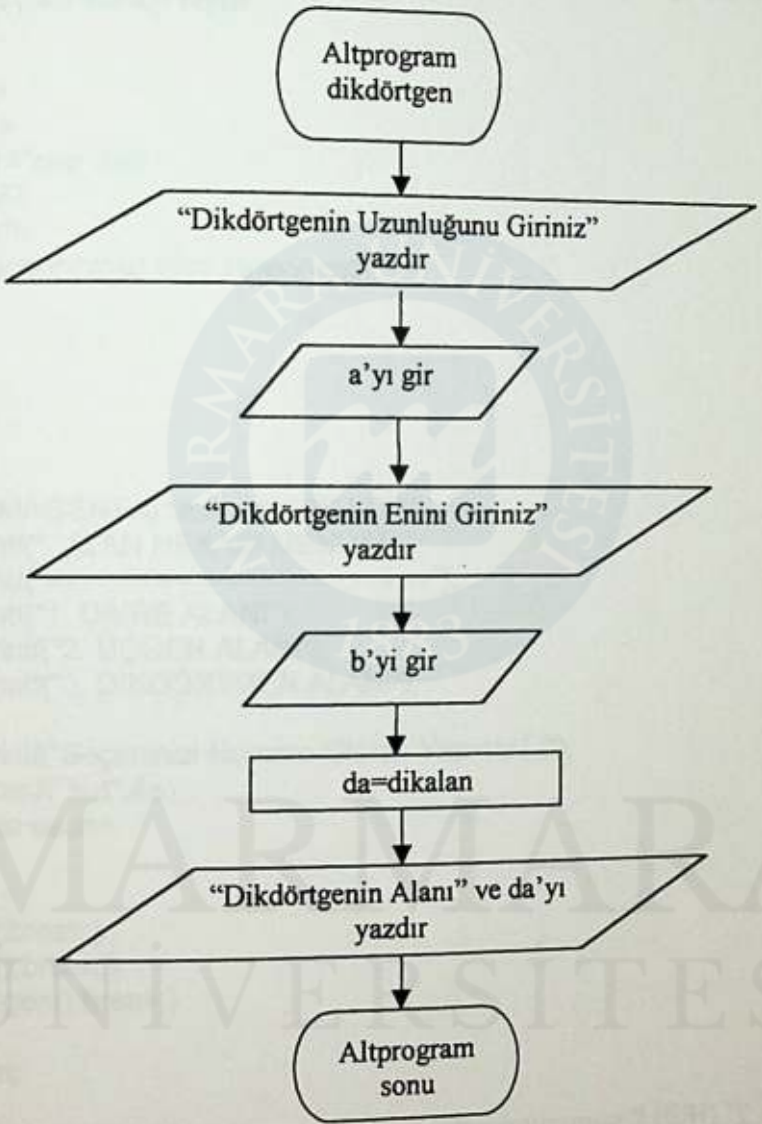
Altprogram ucgen'in akış diyagramı :

MARMARA ÜNİVERSİTESİ YAYINEVİ



Altprogram dikdörtgen'in akış diyagramı :

MARMARA ÜNİVERSİTESİ YAYINEVİ



b) Çizilen program akış diyagramlarına bağlı olarak ve C programlama dili kullanılarak yazılan bilgisayar programı :

MARMARA ÜNİVERSİTESİ YAYINEVİ

Bu program; ekranda görüntülenecek "Alan Hesabı Menüsü" nden kullanıcının yapacağı seçime bağlı olarak, ilgili alan hesapları alt programlarına erişilmesini ve bu alt programlarda girilen değerler itibariyle ilgili geometrik şekilin alan hesabını yaparak, ekrana yazar ve son olarak da menünün son seçeneği ile kullanıcıya programdan çıkış olanağı sağlar.

```
#include<stdio.h>
#include<conio.h>
#define dalan 3.14*cap*cap
#define ualan t*h/2
#define dikalan a*b
void daire(void),ucgen(void),dikdortgen(void);
main()
{
 int s;
 char cevap='e';
 bas:
 clrscr();
 textbackground(MAGENTA);textcolor(YELLOW);
 gotoxy(20,6);printf("  ALAN HESABI MENÜSÜ");
 gotoxy(20,7);printf("======================");
 gotoxy(22,9);printf("1. DAİRE ALANI");
 gotoxy(22,11);printf("2. ÜÇGEN ALANI");
 gotoxy(22,13);printf("3. DİKDÖRTGEN ALANI");
 secim:
 gotoxy(20,16);printf("Seçiminizi Numara Olarak Yapınız [ ]");
 gotoxy(54,16);scanf("%d",&s);
 if (s<1 || s>4) goto secim;
 switch(s)
 {
  case 1 : {daire();break;}
  case 2 : {ucgen();break;}
  case 3 : {dikdortgen();break;}
 }
 if (s==4) goto son;
 tekrar:
 gotoxy(15,15);printf("Başka Hesaplama Yapmak İstiyormusunuz ? (E/H) ");
 cevap=getch();
 if (cevap=='E' || cevap=='e') goto bas;
 son:;
 }
```

```
void daire(void)
{
 float cap,d;
 clrscr();
 gotoxy(15,6);printf("Dairenin Yarıçapını Giriniz ... > ");
 gotoxy(51,6);scanf("%f",&cap);
 d=dalan;
 gotoxy(15,9);printf("Dairenin Alanı ..... : %f",d);
}
void ucgen(void)
{
 float h,t,u;
 clrscr();
 gotoxy(15,6);printf("Üçgenin Yüksekliğini Giriniz ... > ");
 gotoxy(51,6);scanf("%f",&h);
 gotoxy(15,8);printf("Üçgenin Tabanını Giriniz ..... > ");
 gotoxy(51,8);scanf("%f",&t);
 u=ualan;
 gotoxy(15,11);printf("Üçgenin Alanı ..... : %f",u);
  }
 void dikdortgen(void)
 {
  float a,b,da;
  clrscr();
  gotoxy(15,6);printf("Dikdörtgenin Uzunluğunu Giriniz ... > ");
  gotoxy(55,6);scanf("%f",&a);
  gotoxy(15,8);printf("Dikdörtgenin Enini Giriniz ..... > ");
  gotoxy(55,8);scanf("%f",&b);
  da=dikalan;
  gotoxy(15,11);printf("Dikdörtgenin Alanı ..... : %f",da);
 }
```

MARMARA ÜNİVERSİTESİ YAYINEVİ

5.2.3. KARAR TABLOLARI

Karar tablosu incelenen sistemin mantığını, adım adım akış yerine tablo biçiminde gösteren bir araçtır. Karar tabloları, çeşitli koşullar altında gerçekleştirilecek faaliyetleri bir tablo halinde gösterirler. Akış diyagramlarından bağımsız olarak kullanabilecekleri gibi, onları tamamlayan bir yöntem olarak da kullanılabilirler.

Karar tabloları dört bölümden oluşmaktadır. Bir karar tablosunun genel yapısı aşağıda verilmiştir.[15]

I Koşullar Bölümü	II Koşullara İlişkin Çeşitli Durumların Tanımlandığı Bölüm
III Eylemler Bölümü	IV Eylemlerin Kabul veya Reddine İlişkin Bölüm

Örnek 1 :

Bir ticari kuruluş, karşılığı daha sonra ödenmek koşuluyla gelen siparişler hakkında karar vermek istemektedir. Kararların alınmasında standart bir yol izlemek amacıyla, sipariş veren müşteriler dört koşul itibariyle incelenmektedir. Kuruluş, bu koşulların bileşimine (koşulların gerçekleşme durumlarına) bağlı olarak çeşitli eylemler belirlemiştir. Koşullar, çeşitli koşullara ilişkin durumlar, eylemler ve bu eylemlerin kabul veya reddi aşağıda verilen tabloda yer almaktadır.

KOŞULLAR	ÇEŞİTLİ KOŞULLARA İLİŞKİN DURUMLAR											
1. Müşterinin kuruluşumuzla yaptığı eski alış verişlerde borcunu ödeme alışkanlığı iyi mi ?	E	E	E	E	H	H	H	H				
2. Siparişle birlikte borcu kredi limitini aşıyor mu ?	E	E	H	H	E	E	H	H				
3. Kuruluşumuzla ilk defa alış veriş ediyorsa çevresindekilerle olan ilişkilerde şöhreti iyi mi ?									E	E	H	H
4. İyi bir kefili var mı ?	E	H	E	H	E	H	E	H	E	H	E	H
EYLEMLER												
1. Siparişi kabul et.	*		*	*				*	*			
2. Siparişi 1 kereye mahsus kabul et.		*			*		*			*	*	
3. Siparişi reddet.						*						*

[15] Esen, Öner, "Programlamaya Giriş ve Basic", s.33.

Örnek 2 :

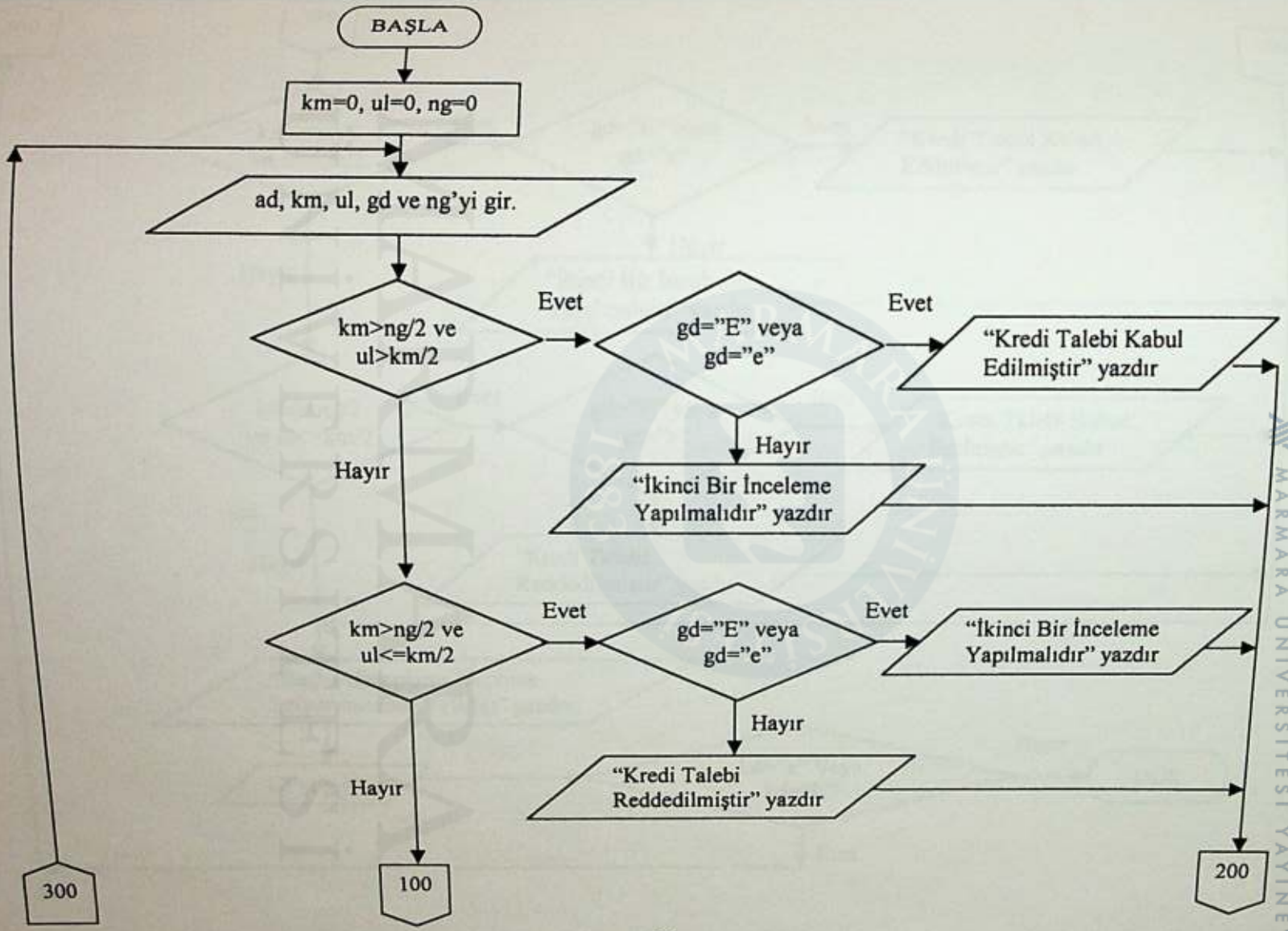
A bankası, müşterilerinin kredi taleplerini cevaplamak amacıyla aşağıda verilen karar tablosunu geliştirmiştir. Bu karar tablosuna bağlı olarak müşteri taleplerini karara bağlayacak çalışmaların bilgisayar desteğinde yürütülmesi istenmektedir.[16]

KOŞULLAR	1	2	3	4	5	6	7	8
İstenen Kredi Miktarı > (Yıllık Net Gelir) / 2	E	E	E	E	H	H	H	H
Banka Hesabı Üst Limiti > (İstenen Kredi Miktarı) / 2	E	E	H	H	E	E	H	H
Gayri Menkulü Var mı?	E	H	E	H	E	H	E	H
KARARLAR								
İstenen Krediyi Ver	*				*		*	
İkinci Bir İnceleme Yap		*	*			*		
Kredi Talebini Reddet				*				*

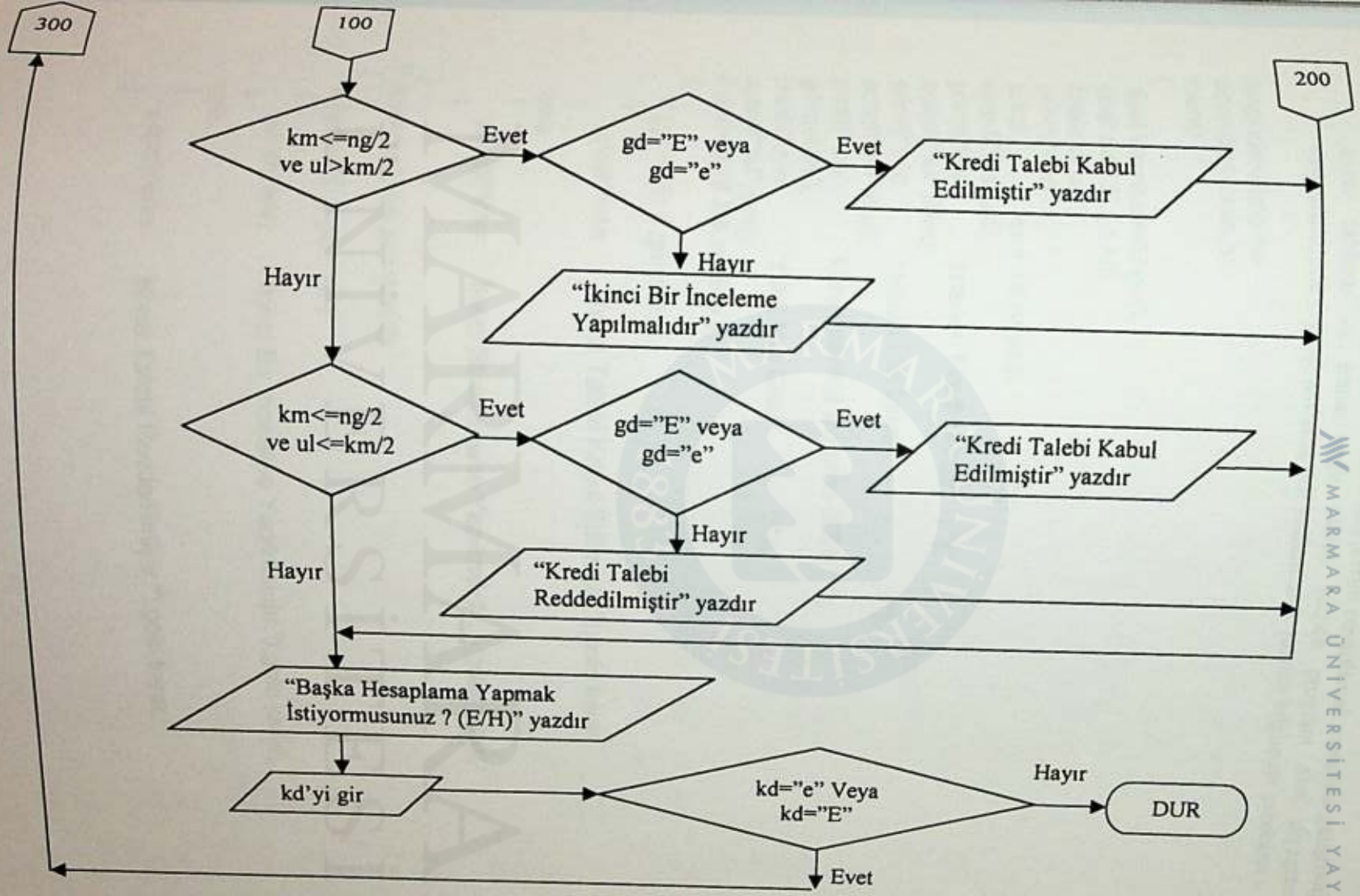
a. Yukarıda verilen karar tablosuna bağlı olarak ve yazılacak bilgisayar programına temel olacak program akış diyagramını çiziniz.

b. Çizilen akış diyagramlarına bağlı olarak ve C programlama dilini kullanarak gerekli olan bilgisayar programını yazınız.

[16] ibid s.144.



MARMARA ÜNİVERSİTESİ YAYINEVİ



MARMARA ÜNİVERSİTESİ YAYINEVİ

b) A bankasının müşterilerinin kredi taleplerini cevaplamak amacıyla geliştirdiği karar tablosu ve buna bağlı olarak çizilen program akış diyagramı doğrultusunda C programlama dili kullanılarak yazılan bilgisayar programı :

MARMARA ÜNİVERSİTESİ YAYINEVİ

```
#include<stdio.h>
#include<conio.h>
main()
{
 float km=0,ul=0,ng=0;
 char ad[15],gd,kd;
 basla:clrscr();
 printf("\n\n\n");
 printf("       İsim ve Soyadı          :");
 scanf("%s",ad);
 printf("\n\n       İstenen Kredi Miktarı  : ");
 scanf("%f",&km);
 printf("\n\n       Hesap Üst Limiti              : ");
 scanf("%f",&ul);
 printf("\n\n       Gayrimenkul Durumu (E/H) : ");
 gd=getch();
 printf("\n\n       Yıllık Net Gelir        : ");
 scanf("%f",&ng);
 if (km>ng/2 && ul>km/2)
 {
   if (gd=='E' || gd=='e')
   {
     printf("\n\n      Kredi Talebi Kabul Edilmiştir.");goto karar;
   }
   else
   {
     printf("\n\n      İkinci Bir İnceleme Yapılmalıdır.");goto karar;
   }
 }
 if (km>ng/2 && ul<=km/2)
 {
   if (gd=='E' || gd=='e')
   {
     printf("\n\n      İkinci Bir İnceleme Yapılmalıdır.");goto karar;
   }
   else
   {
     printf("\n\n      Kredi Talebi Reddedilmiştir.");goto karar;
   }
 }
```

```
if (km<=ng/2 && ul>km/2)
{
  if (gd=='E' || gd=='e')
   {
     printf("\n\n      Kredi Talebi Kabul Edilmiştir.");goto karar;
   }
  else
   {
     printf("\n\n      İkinci Bir İnceleme Yapılmalıdır.");goto karar;
   }
}
if (km<=ng/2 && ul<=km/2)
{
  if (gd=='E' || gd=='e')
   {
    printf("\n\n          Kredi Talebi Kabul Edilmiştir.");goto karar;
   }
  else
   {
    printf("\n\n          Kredi Talebi Reddedilmiştir.");goto karar;
   }
}
karar:
printf("\n\n\n      Baska Hesaplama Yapmak İstiyormusunuz ? (E/H) ");
kd=getch();
if (kd=='E' || kd=='e') goto basla;
clrscr();
}
```

MARMARA ÜNİVERSİTESİ YAYINEVİ

6. C PROGRAMLAMA DİLİ İLE BİR DOSYALAMA VE İSTATİSTİKSEL ANALİZ UYGULAMASI

"Sistem Analizi – C Programlama Dili Örnekleriyle" isimli çalışmamızın son bölümünde, temel dosyalama işlemleriyle birlikte, istatistiksel analiz içeren ve C programlama dili kullanılarak geliştirilen bir bilgisayar programı sunulmaktadır.

6.1. PROGRAMIN KONUSU VE AMACI

Bu program kullanıcı tarafından ismi klavyeden girilen veri dosyasının yaratılmasını sağlayarak başlamaktadır. Bu veri dosyasında, üniversite öğrencilerine ait temel bilgiler ve aldıkları iki dersle ilgili olarak vize, final ve ortalama notları kayıtlar halinde saklanmaktadır. Programın amacı üniversite öğrencileri ile ilgili bir veri tabanı yaratılması ve bu veri tabanının üstünde çeşitli istatistiksel analizler gerçekleştirilmesidir. Bu amaçla yapılan çalışma, sınırlı sayıda değişken kullanılarak bir deneme mahiyetinde yapılmış olup, yapılacak yeni çalışmalar ile geliştirilmesi tasarlanmıştır.

Program, yeni yaratılan veya daha önce yaratılmış olan veri dosyasına yeni kayıt girişi, bu veri dosyasında isim ve soyadı itibariyle kayıt arama, tüm kayıtların dökümü, mevcut kayıtlar üzerinde düzeltme yapılması, veri dosyası kayıtlarından arzu edilenin silinmesi ve mevcut kayıtlar üzerinde çeşitli istatistiksel analizler yapılmasını sağlamaktadır.

İstatistiksel analizler de sınırlı tutulmuş olup, ilk olarak eleman sayısı, minimum, maksimum, ortalama ve standart sapma gibi istatistiksel ölçüler hesaplanmaktadır. Bunları takiben öğrencilere ait başarı notları arasındaki korelasyon, determinasyon katsayılarının ve regresyon denklemlerinin hesaplanmasını kapsayan bir istatistiksel ilişki analizi gerçekleştirilmektedir.

6.2. PROGRAMIN KAPSAMI

Geliştirilen bu C programı çalıştırıldığı zaman, kullanıcı

"Veri Dosyası İsmini Giriniz . . . > "

şeklinde bir mesaj ile karşılaşmaktadır. Bu mesajı takiben kullanıcının, yaratılacak yeni veri dosyasının ismini veya daha önce yaratılmış olan bir veri dosyasının ismini girmesi gerekmektedir.

Girilen isme sahip bir veri dosyası yok ise, program bu isimle yeni bir veri dosyası yaratmaktadır. Eğer girilen isme sahip bir veri dosyası var ise, program bu veri dosyasını kullanıma açmaktadır.

Bu aşamayı takiben aşağıda verilen menü ekranda görüntülenmektedir.

VERİ ANALİZİ
(Y) eni Kayıt Girişi Kayıt (A) rama Kayıt (D) üzeltme Kayıt (S) ilme (T) üm Kayıtların Dökümü
İstatistiksel Analiz (z)
Çı (k) ış

MARMARA ÜNİVERSİTESİ YAYINEVİ

Seçiminizi Yapınız ...

6.2.1. YENİ KAYIT GİRİŞİ

"Seçiminizi Yapınız" şeklindeki mesajı takiben "Y" veya "y" tuşuna basılması halinde yeni veya daha önce yaratılmış olan bir veri dosyasına, yeni kayıt girişi yapılabilmektedir. Yeni kayıt girişi şu verilerden oluşmaktadır.

- İsim
- Soyadı
- Doğum Yeri
- Doğum Yılı
- Uyruğu
- Cinsiyeti
- Ö.S.Y.M. Puanı
- 1. Dersin Adı
- Vize Notu
- Final Notu

* Sınav notlarının girişini takiben 1. Dersin Başarı Notu hesaplanarak, görüntülenmektedir.

- 2. Dersin Adı
- Vize Notu
- Final Notu

* Sınav notlarının girişini takiben 2. Dersin Başarı Notu hesaplanarak, görüntülenmektedir.

Veri girişinin sonunda "Kaydetmek İstiyor musunuz? (E/H)" şeklindeki soru ekranda görüntülenmekte, kullanıcının "E" veya "e" girmesi ile, girişi yapılan veriler yeni kayıt olarak veri dosyasına kaydedilmektedir. Diğer halde ise bu veriler dosyaya kayıt edilmemektedir.

Yeni kayıt girişi sürecinin son aşamasında "Başka Kayıt Yapacak mısınız ? (E/H)" sorusu görüntülenir. Kullanıcının "E" veya "e" tuşuna basması halinde, veri dosyasına yeni bir kayıt daha girilmekte, diğer halde ise "Veri Analizi" isimli menüye geri dönülmektedir.

6.2.2. KAYIT ARAMA

Eğer "Veri Analizi" isimli menüden seçim "A" veya "a" girilerek yapılırsa, kullanıcı ekranda birbirini takip eden iki mesajla karşılaşır.

"Aradığınız Kaydın İsmini Giriniz . . . . . >"

"Aradığınız Kaydın Soyadını Giriniz . . . >"

Aranılan kaydın isim ve soyadı kullanıcı tarafından girilir. Eğer aranılan kayıt bulunursa, kayıta ait veriler ekranda görüntülenir. Aranılan kayıtın bulunamaması halinde ise, "Aradığınız Kayıt Bulunamamıştır." şeklindeki mesaj ile kullanıcıya bilgi verilir.

Her iki durumda da son olarak "Menüye Dönüş İçin Enter Tuşuna Basınız." Mesajı görüntülenmektedir. Kullanıcının Enter tuşuna basması ile "Veri Analizi" isimli menüye geri dönülür.

6.2.3. KAYIT DÜZELTME

Eğer "Veri Analizi" isimli menüden seçim "D" veya "d" girilerek yapılırsa, kullanıcı aşağıda verilen mesajlarla karşılaşır.

"Düzeltilecek Kayıtın"

"İsmini Giriniz . . . . . >"

"Soyadını Giriniz . . . >"

Düzeltilecek kayıtın isim ve soyadı kullanıcı tarafından girilir. Eğer aranılan kayıt bulunursa, kayıta ait veriler ekranda görüntülenir ve görüntülenen verilerin altında "Bu Kayıtta Düzeltme Yapmak İstiyor musunuz? (E/H)" şeklindeki mesaj görüntülenir. Kullanıcının "E" veya "e" tuşlarına basması ile kayıt üzerinde düzeltme süreci başlar. Kayıtta düzeltmeler yapıldıktan sonra "Menüye Dönüş İçin Enter Tuşuna Basınız." şeklindeki mesaja bağlı olarak, Enter tuşuna basıldığında "Veri Analizi" isimli menüye geri dönülür.

Aranılan kayıtın bulunamaması halinde ise "Aradığınız Kayıt Bulunamamıştır." ve "Menüye Dönüş İçin Enter Tuşuna Basınız." şeklindeki mesajlar görüntülenir. Kullanıcının Enter tuşuna basması ile "Veri Analizi" isimli menüye geri dönülür.

6.2.4. KAYIT SİLME

Eğer "Veri Analizi" isimli menüden seçim "S" veya "s" girilerek yapılırsa, kullanıcı aşağıda verilen mesajlarla karşılaşır.

"Silinecek Kayıtın"

"İsmini Giriniz . . . . . >"

"Soyadını Giriniz . . . >"

Silinecek kayıtın isim ve soyadı kullanıcı tarafından girilir. Eğer silinecek kayıt bulunursa, kayıta ait veriler ekranda görüntülenir ve "Bu Kaydın Silinmesini İstiyor musunuz? (E/H)"

şeklindeki mesaja bağlı olarak "E" veya "e" tuşuna basılırsa, görüntülenen kayıt veri dosyasından silinir. Diğer halde ise kayıt veri dosyasından silinmez. Aranılan kayıtın bulunamaması halinde ise "Aradığınız Kayıt Bulunamamıştır." şeklindeki mesaj ile kullanıcıya bilgi verilir.

Her iki durumda da son olarak "Menüye Dönüş İçin Enter Tuşuna Basınız." Mesajı görüntülenmektedir. Kullanıcının Enter tuşuna basması ile "Veri Analizi" isimli menüye geri dönülmektedir.

6.2.5. TÜM KAYITLARIN DÖKÜMÜ

Eğer "Veri Analizi" isimli menüden seçim "T" veya "t" girilerek yapılırsa, veri dosyasına ait kayıtlar birer, birer görüntülenir. Her kayıt görüntülendikten sonra, "Sonraki Kayıt İçin Enter Tuşuna Basınız." mesajı görüntülenir. Enter tuşuna bastıkça bir sonraki kayıt ekranda görüntülenir. Son kayıt ta görüntülendikten sonra "Veri Analizi" isimli menüye geri dönülür.

6.2.6. İSTATİSTİKSEL ANALİZ

Tüm kayıtlarda iki ayrı ders için vize ve final notları girildikten sonra başarı notları yani ortalamalar hesaplanmaktadır.

İstatistiksel Analiz seçeneği ile program, bu derslerin başarı notlarına ilişkin olarak aşağıdaki analiz sonuçlarını bir rapor şeklinde sunmaktadır:

- Öğrenci Sayısı { Veri Dosyasındaki Toplam Kayıt Sayısı }

- ..... Dersindeki Minimum Başarı Notu
- ..... Dersindeki Maksimum Başarı Notu
- ..... Dersindeki Ortalama Başarı Notu
- ..... Dersindeki Başarı Notları Standart Sapması

{ Birinci Derse Ait Başarı Notlarına İlişkin Olarak Yapılan İstatistiksel Analiz Sonuçları }

- ..... Dersindeki Minimum Başarı Notu
- ..... Dersindeki Maksimum Başarı Notu
- ..... Dersindeki Ortalama Başarı Notu
- ..... Dersindeki Başarı Notları Standart Sapması

{ İkinci Derse Ait Başarı Notlarına İlişkin Olarak Yapılan İstatistiksel Analiz Sonuçları }

- ..... ve ..... Derslerine Ait Başarı Notları Arasındaki Korelasyon Katsayısı .....

- ..... ve ..... Derslerine Ait Başarı Notları Arasındaki Determinasyon Katsayısı .....

- Birinci Regresyon Denklemi
- İkinci Regresyon Denklemi

İstatistiksel Analiz sonuçlarını takiben "Menü İçin Enter Tuşuna Basınız." mesajı görüntülenmektedir. Bu mesaja bağlı olarak, Enter tuşuna basıldığında "Veri Analizi" menüsüne geri dönülür.

6.2.7. ÇIKIŞ

"Veri Analizi" menüsünün son seçeneği "Çıkış" olup, bu seçenek programın çalışmasını sona erdirmektedir. Bunun için "K" veya "k" tuşuna basılması yeterli olmaktadır.

6.3. PROGRAM METNİ

```
/* Kullanıcı Tarafından İsmi Girilen Veri Dosyasının Yaratılması, */
/* Bu Veri Dosyası Üzerinde Temel Dosyalama İşlemlerinin Gerçekleştirilmesi ve */
/* Veriler Üstünde İstatistiksel Analizler Yapılması */

#include <stdio.h>
#include <ctype.h>
#include <string.h>
#include <stdlib.h>
#include <conio.h>
#include <math.h>
#define  SIZE 100
        struct talebe {
        char ad[15];
        char soy[15];
        char dogyer[20];
        int dogyil;
        char uyruk[10];
        char cinsi[10];
        float osym;
        char ders1[10];
        int vize1;
        int final1;
        float orta1;
        char ders2[10];
        int vize2;
        int final2;
        float orta2;
        };
struct talebe tal[SIZE];

void initlist(void), tum(void);
void save(void), load(void), ara(void), goster(int t);
void yeni(void), duzelt(void), silme(void), istat(void);
menu(void);
void kutu(int x1,int y1, int x2, int y2);
char dosyaadi[81];

main()
{
  char sec;
/* Yaratılacak veya Kullanılacak Veri Dosyasının İsmi */
/* Kullanıcı  Tarafından Giriliyor. */
```

```
clrscr();
gotoxy(15,7);printf("Veri Dosyası İsmini Giriniz . . . > ");
gets(dosyaadi);

initlist();
for(;;)
  {  sec=menu();
       switch(sec)
       {
         case 'y': yeni();
              break;
         case 'a': ara();
              break;
         case 'd': duzelt();
              break;
         case 's': silme();
              break;
         case 't': tum();
              break;
         case 'z': istat();
              break;
         case 'k': clrscr(); exit(1);
       }
   }
}

/*tal isimli dizininin initialize edilmesi */
void initlist(void)
{
  register int t;

  for(t=0;t<SIZE;t++)
       *tal[t].ad='\0';
}
/* Menüden Seçim Yapma */
menu(void)
{
  char s[80];
  textbackground(BLUE);
  do
   { clrscr();
     textcolor(YELLOW);
     kutu(24,3,54,15);
     textbackground(MAGENTA);
     gotoxy(26,4);cprintf("       VERİ ANALİZİ      ");
     textbackground(BLUE);
     gotoxy(24,5);cprintf(" ==============================|");
     gotoxy(26,6);cprintf(" (Y)eni Kayıt Girişi");
     gotoxy(26,7);cprintf(" Kayıt (A)rama");
     gotoxy(26,8);cprintf(" Kayıt (D)üzeltme");
```

MARMARA ÜNİVERSİTESİ YAYINEVİ

```
    gotoxy(26,9);cprintf("  Kayıt (S)ilme");
    gotoxy(26,10);cprintf("  (T)üm Kayıtların Dökümü");
    gotoxy(24,11);cprintf("  ================================|");
    gotoxy(26,12);cprintf("  İstatistiksel Anali(z)");
    gotoxy(24,13);cprintf("  ================================|");
    gotoxy(26,14);cprintf("  Çı(k)ış");
    gotoxy(26,16);cprintf("  Seçiminizi yapınız ...");
    gets(s);
  } while (!strchr("yadstkz", tolower(*s)));
  return tolower(*s);
}
void yeni(void)
{
  register int i;
  char ch,osy[81];
  clrscr();
  load();
  for(i=0; i<SIZE; i++)
      if(!*tal[i].ad)
        break;
        if(i==SIZE) /* Kayıtlar Doldu */
          {printf("Kayıtlar Dolmuştur!\n");
           return;
}
/* Yeni Bilgi Girişi */
    do
      {
      clrscr();
      gotoxy(20,2);printf("İsim                          : ");
      scanf("%s",tal[i].ad);
      gotoxy(20,3);printf("Soyadı                        : ");
      scanf("%s",tal[i].soy);
      gotoxy(20,4);printf("Doğum Yeri                    : ");
      scanf("%s",tal[i].dogyer);
      gotoxy(20,5);printf("Doğum Yılı                    : ");
      scanf("%d",&tal[i].dogyil);
      gotoxy(20,6);printf("Uyruğu                        : ");
      scanf("%s",tal[i].uyruk);
      gotoxy(20,7);printf("Cinsiyeti                     : ");
      scanf("%s",tal[i].cinsi);
      gotoxy(20,8);printf("Ö.S.Y.M Puanı (493.55)         : ");
      scanf("%s",osy);
      tal[i].osym=atof(osy);
      gotoxy(20,10);printf("1. Dersin Adı                 : ");
      scanf("%s",tal[i].ders1);
      gotoxy(20,11);printf("Vize Notu                     : ");
      scanf("%d",&tal[i].vize1);
      gotoxy(20,12);printf("Final Notu                    : ");
      scanf("%d",&tal[i].final1);
      tal[i].orta1=0.4*tal[i].vize1+0.6*tal[i].final1;
```

```
      gotoxy(20,13);printf("Başarı Notu                    : %.0f",tal[i].orta1);
      gotoxy(20,15);printf("2. Dersin Adı                  : ");
      scanf("%s",tal[i].ders2);
      gotoxy(20,16);printf("Vize Notu                      : ");
      scanf("%d",&tal[i].vize2);
      gotoxy(20,17);printf("Final Notu                     : ");
      scanf("%d",&tal[i].final2);
      tal[i].orta2=0.4*tal[i].vize2+0.6*tal[i].final2;
      gotoxy(20,18);printf("Başarı Notu                    : %.0f",tal[i].orta2);
      gotoxy(20,20);printf("Kaydetmek İstiyor musunuz ? (E/H)  ");
      do
       {
        ch=tolower(getch());
       }
      while (ch != 'e' && ch != 'h');

       if(ch == 'e')
         save();
      gotoxy(20,22);printf("Başka Kayıt Yapacak mısınız ? (E/H)");
      ch=tolower(getch());
      if( ch== 'e') i++;
     }
    while( ch == 'e');
}
/* Tüm Kayıtların Dökümü */
void tum(void)
{
  register int t;
  load();
  for(t=0;t<SIZE;t++)
  {
   if(*tal[t].ad)
   {
    goster(t);
    gotoxy(15,23);printf("Sonraki Kayıt İçin Herhangi Bir Tuşa Basınız...");
    getch();
   }
  }
}
/* Dosyaya Kaydetme */
void save(void)
{
  FILE *fp;
  register int i;

 if((fp=fopen(dosyaadi,"wb"))==NULL)
   {
    printf("%s Dosyası Açılamıyor\n",dosyaadi);
    return;
   }
```

MARMARA ÜNİVERSİTESİ YAYINEVİ

```
for(i=0;i<SIZE;i++)
  if(*tal[i].ad)
     if(fwrite(&tal[i],sizeof(struct talebe),1,fp)!=1)
       printf("Dosyaya Yazılamıyor.\n");
     fclose(fp);

}
/* Dosyadan Yükleme */
void load(void)
{
  FILE *fp;
  register int i;
  if((fp=fopen(dosyaadi,"rb"))==NULL)
    {
      printf("Dosya Açılamıyor.\n");
      return;
    }
  initlist();
  for(i=0;i<SIZE;i++)
    if(fread(&tal[i],sizeof(struct talebe),1,fp)!=1)
     {
       if(feof(fp))
       {
         fclose(fp);
         return;
       }
       printf("Dosya Okuma Hatası\n");
     }
}

void kutu(int x1,int y1, int x2, int y2)
       {int i;
        gotoxy(x1,y1);putch(201);
        gotoxy(x2,y1);putch(187);
        gotoxy(x1,y2);putch(200);
        gotoxy(x2,y2);putch(188);
        for(i=x1+1;i<x2;i++) {gotoxy(i,y1);putch(205);}
        for(i=x1+1;i<x2;i++) {gotoxy(i,y2);putch(205);}
        for(i=y1+1;i<y2;i++) {gotoxy(x1,i);putch(186);}
        for(i=y1+1;i<y2;i++) {gotoxy(x2,i);putch(186);}
       }

/* Klavyeden Girilen İsim ve Soyadı İtibariyle Kayıt Arama */
void ara(void)
{
  register int i;
  char ch, adi[15],soyadi[15];
  clrscr();
  load();
  gotoxy(10,5);printf("Aradığınız Kaydın İsmini Giriniz ..... > ");
  gets(adi);
```

MARMARA ÜNİVERSİTESİ YAYINEVİ

```
  gotoxy(10,8);printf("Aradığınız Kaydın Soyadını Giriniz ..... > ");
  gets(soyadi);

  for(i=0; i<SIZE; i++)    /* İsmi ve Soyadı Uyan İlk Kayıt Bulunuyor */
      {
        if(strcmp(tal[i].ad,adi)==0 && strcmp(tal[i].soy,soyadi)==0)  break;
      }
        if(i==SIZE)
          {gotoxy(10,12);printf("Aradığınız Kayıt Bulunamamıştır");
           gotoxy(10,15);printf("Menüye Dönüş İçin Enter Tuşuna Basınız.");getch();
           return;
          }
  goster(i);
  gotoxy(15,23);printf("Menüye Dönüş İçin Enter Tuşuna Basınız.");getch();
}
/* Bulunan Kayıdın Görüntülenmesi */
void goster(int t)
{
 clrscr();
       gotoxy(15,4);printf("İsim                 : %s",tal[t].ad);
       gotoxy(15,4);printf("Soyadı               : %s",tal[t].soy);
       gotoxy(15,4);printf("Doğum Yeri           : %s",tal[t].dogyer);
       gotoxy(15,4);printf("Doğum Yılı           : %d",tal[t].dogyil);
       gotoxy(15,4);printf("Uyruğu               : %s",tal[t].uyruk);
       gotoxy(15,4);printf("Cinsiyeti            : %s",tal[t].cinsi);
       gotoxy(15,4);printf("Ö.S.Y.M Puanı        : %7.3f",tal[t].osym);
       gotoxy(15,4);printf("1. Dersin Adı        : %s",tal[t].ders1);
       gotoxy(15,4);printf("Vize Notu            : %d",tal[t].vize1);
       gotoxy(15,4);printf("Final Notu           : %d",tal[t].final1);
       gotoxy(15,4);printf("Başarı Notu          : %.0f",tal[t].orta1);
       gotoxy(15,4);printf("2. Dersin Adı        : %s",tal[t].ders2);
       gotoxy(15,4);printf("Vize Notu            : %d",tal[t].vize2);
       gotoxy(15,4);printf("Final Notu           : %d",tal[t].final2);
       gotoxy(15,4);printf("Başarı Notu          : %.0f",tal[t].orta2);
}
/* Kayıt Silme Programı */
void silme(void)
{
  register int i, t;
  char ch, adi[15],soyadi[15];
  clrscr();
  load();
  gotoxy(20,5);printf("Silinecek Kayıtın");
  gotoxy(20,7);printf("İsmini Giriniz ..... > ");
  gets(adi);
  gotoxy(20,9);printf("Soyadını Giriniz ..... > ");
  gets(soyadi);
```

MARMARA ÜNİVERSİTESİ YAYINEVİ

```
for(i=0; i<SIZE; i++)    /* İsim ve Soyadı Uyan İlk Kayıt Bulunuyor */
    {
     if(strcmp(tal[i].ad,adi)==0 && strcmp(tal[i].soy,soyadi)==0)  break;
    }
      if(i==SIZE)
        {gotoxy(20,12);printf("Aradığınız Kayıt Bulunamamıştır.");
         gotoxy(20,15);printf("Menüye Dönüş İçin Enter Tuşuna Basınız.");getch();
         return;
        }
goster(i);
gotoxy(15,23);printf("Bu Kaydın Silinmesini İstiyor musunuz ? (E/H)");
gotoxy(58,23);
     do
      {
       ch=tolower(getche());
      } while (ch != 'e' && ch != 'h');

      if(ch == 'e')
        for(t=i; t<SIZE-1; t++)
            {
             strcpy(tal[t].ad,tal[t+1].ad);
             strcpy(tal[t].soy,tal[t+1].soy);
             strcpy(tal[t].dogyer,tal[t+1].dogyer);
             tal[t].dogyil=tal[t+1].dogyil;
             strcpy(tal[t].uyruk,tal[t+1].uyruk);
             strcpy(tal[t].cinsi,tal[t+1].cinsi);
             tal[t].osym=tal[t+1].osym;
             strcpy(tal[t].ders1,tal[t+1].ders1);
             tal[t].vize1=tal[t+1].vize1;
             tal[t].final1=tal[t+1].final1;
             tal[t].orta1=tal[t+1].orta1;
             strcpy(tal[t].ders2,tal[t+1].ders2);
             tal[t].vize2=tal[t+1].vize2;
             tal[t].final2=tal[t+1].final2;
             tal[t].orta2=tal[t+1].orta2;
            }
      save();
}
/* Kayıtlar Üstünde Düzeltmeler Yapılmasını Sağlayan Program */
void duzelt(void)
{
 register int i, t;
 char ch, adi[15],soyadi[15];
 clrscr();
 load();
 gotoxy(20,4);printf("Düzeltilecek Kayıtın");
 gotoxy(20,7);printf("İsmini Giriniz        : ");
 gets(adi);
 gotoxy(20,9);printf("Soyadını giriniz      : ");
 gets(soyadi);
```

```
for(i=0; i<SIZE; i++)     /* Adı ve Soyadı Uyan İlk Kayıt Bulunuyor */
    {
     if(strcmp(tal[i].ad,adi)==0 && strcmp(tal[i].soy,soyadi)==0)  break;
    }
     if(i==SIZE)
       {gotoxy(20,12);printf("Aradığınız Kayıt Bulunamamıştır.");
        gotoxy(20,16);printf("Menüye Dönüş İçin Enter Tuşuna Basınız.");getch();
        return;
       }
goster(i);
gotoxy(15,23);printf("Bu Kayıtta Düzeltme Yapmak İstiyor musunuz ? (E/H)");
gotoxy(63,23);
    do
     {
      ch=tolower(getche());
     } while (ch != 'e' && ch != 'h');

     if(ch == 'e')
          {
           clrscr();
           gotoxy(15,3);printf("İsim                : ");
           gets(tal[i].ad);
           gotoxy(15,4);printf("Soyadı              : ");
           gets(tal[i].soy);
           gotoxy(15,5);printf("Doğum Yeri          : ");
           gets(tal[i].dogyer);
           gotoxy(15,6);printf("Doğum Yılı          : ");
           scanf("%d",&tal[i].dogyil);
           gotoxy(15,7);printf("Uyruğu              : ");
           scanf("%s",tal[i].uyruk);
           gotoxy(15,8);printf("Cinsiyeti           : ");
           scanf("%s",tal[i].cinsi);
           gotoxy(15,9);printf("Ö.S.Y.M Puanı        : ");
           scanf("%f",&tal[i].osym);
           gotoxy(15,10);printf("1. Dersin Adı       : ");
           scanf("%s",tal[i].ders1);
           gotoxy(15,11);printf("Vize Notu           : ");
           scanf("%d",&tal[i].vize1);
           gotoxy(15,12);printf("Final Notu          : ");
           scanf("%d",&tal[i].final1);
           tal[i].orta1=0.4*tal[i].vize1+0.6*tal[i].final1;
           gotoxy(15,13);printf("Başarı Notu         : %.0f",tal[i].orta1);
           gotoxy(15,14);printf("2. Dersin Adı       : ");
           scanf("%s",tal[i].ders2);
           gotoxy(15,15);printf("Vize Notu           : ");
           scanf("%d",&tal[i].vize2);
           gotoxy(15,16);printf("Final Notu          : ");
           scanf("%d",&tal[i].final2);
           tal[i].orta2=0.4*tal[i].vize2+0.6*tal[i].final2;
           gotoxy(15,17);printf("Başarı Notu         : %.0f",tal[i].orta2);
```

```
            gotoxy(15,20);printf("Herhangi Bir Tuşa Basınız...");
            getch();
            save();
           }
}
/* İstatistiksel Analiz */
void istat(void)
{
 register int t;
 int adet=0;
 float min1=101, min2=101, max1=-1, max2=-1, sum1=0, sum2=0;
 float ortal1, ortal2, sapmak12=0, sapmak1=0, sapmak2=0, sts1, sts2, r;
 float a1, b1, a2, b2;
 load();
 for(t=0;t<SIZE;t++)
 {
  if(*tal[t].ad)
  {
     adet++;
     if(tal[t].orta1 < min1)          min1 = tal[t].orta1;
     if(tal[t].orta2 < min2)          min2 = tal[t].orta2;
     if(tal[t].orta1 > max1)          max1 = tal[t].orta1;
     if(tal[t].orta2 > max2)          max2 = tal[t].orta2;
     sum1=sum1+tal[t].orta1;
     sum2=sum2+tal[t].orta2;
  }
 }
  ortal1=sum1/adet;
  ortal2=sum2/adet;
 for(t=0;t<adet;t++)
 {
     sapmak1=sapmak1+(tal[t].orta1-ortal1)*(tal[t].orta1-ortal1);
     sapmak2=sapmak2+(tal[t].orta2-ortal2)*(tal[t].orta2-ortal2);
     sapmak12= sapmak12+(tal[t].orta1-ortal1)*(tal[t].orta2-ortal2);
 }
  sts1=sqrt(sapmak1/(adet-1));
  sts2=sqrt(sapmak2/(adet-1));
  r=sapmak12/(sqrt(sapmak1*sapmak2));
  b1 = sapmak12/sapmak1;
  b2 = sapmak12/sapmak2;
  a1 = ortal2-b1*ortal1;
  a2 = ortal1-b2*ortal2;
  clrscr();
  printf("\n     Öğrenci Sayısı ..... : %d\n",adet);
  printf("\n     %s Dersindeki Minimum Başarı Notu ..... : %0.f", tal[0].ders1, min1);
  printf("\n     %s Dersindeki Maksimum Başarı Notu ..... : %0.f", tal[0].ders1, max1);
  printf("\n     %s Dersindeki Ortalama Başarı Notu ..... : %0.f", tal[0].ders1, ortal1);
  printf("\n     %s Dersindeki Başarı Notları Standart Sapması : %.2f", tal[0].ders1, sts1 );
  printf("\n\n     %s Dersindeki Minimum Başarı Notu ..... : %0.f", tal[0].ders2, min2);
  printf("\n     %s Dersindeki Maksimum Başarı Notu ..... : %0.f", tal[0].ders2, max2);
```

MARMARA ÜNİVERSİTESİ YAYINEVİ

```
    printf("\n     %s Dersindeki Ortalama Başarı Notu .....  : %0.f", tal[0].ders2, ortal2);
    printf("\n     %s Dersindeki Başarı Notları Standart Sapması : %.2f", tal[0].ders2, sts2 );
    printf("\n\n     %s ve %s Derslerine Ait Başarı Notları Arasındaki\n     Korelasyon
Katsayısı ..... : %.2f", tal[0].ders1, tal[0].ders2, r);
    printf("\n\n     %s ve %s Derslerine Ait Başarı Notları Arasındaki\n
Determinasyon Katsayısı ..... : %.2f", tal[0].ders1, tal[0].ders2, r*r);
    printf("\n\n     Birinci Regresyon Denklemi ..... :");
    printf("  y = %.2f + %.2fx",a1,b1);
    printf("\n     İkinci Regresyon Denklemi ..... :");
    printf("  x = %.2f + %.2fy",a2,b2);
    printf("\n\n     Menüye Dönüş İçin Enter Tuşuna Basınız. ");
    getch();
}
```

MARMARA ÜNİVERSİTESİ 1883

7. SONUÇ

MARMARA ÜNİVERSİTESİ YAYINEVİ

Evrensel sistemin bir alt sistemi olan dünyamız sonsuz sayıda alt sistemden oluşur. Bu alt sistemlerden bir tanesi de insandır. İnsan yaşayan bir sistem olarak yine bir çok alt sistemin bütünüdür. Günümüzde insan dünyamızın sınırları dışına çıkarak, yaratıldığı dünyadan çok evrenin bir alt sistemi niteliğine kavuşmuştur. Kendi dışında kalan sistemlerle etkileşim içindedir. Bu sistemlerden olumlu ve olumsuz yönde etkilenirken, kendisi, daha büyük ölçekte bu sistemleri olumlu ve olumsuz yönde etkilemektedir. İnsanoğlu yaratıldığı evrensel sistem içinde, kendi dışında kalan tüm sistemleri ve şüphesiz kendi alt sistemlerini incelemek, anlamak ve çözmek yolunda sonsuz bir uğraş ve çaba içinde olmuştur. Bu arada çağlar boyunca kendini geliştirmiş ve geliştirmeye devam etmektedir. İnsanın yarattığı uygarlık, var olan evrensel sistemi ve onun alt sistemlerini inceleyerek çözdüğü ve kendi yarattığı sistemler yoluyla katkıda bulunduğu ölçüde gelişmiştir. Bu şekilde de gelişmeye devam edecektir.

İnsanın kendi alt sistemlerini ve çevresindeki tüm sistemleri incelerken ve bu sistemleri anlamaya çalışırken temel amacı kendi sonsuz ihtiyaçlarını tatmin etmek olmuştur. Bu ihtiyaç, kendi merakını giderme ihtiyacı olabileceği gibi, kendi yaşamına olumsuz etkileri olan sistemlerin etkilerine giderme ihtiyacı veya daha rahat, daha iyi ve daha güvenli yaşamak şeklindeki bir ihtiyaç da olabilir.

Bilinçli veya bilinçsiz, insanoğlu yaratıldığı günden bu güne hep analizci olmuştur. Var olan sistemleri analiz ederken ve bunlardan yararlanırken yetinmemiş, kendi sonsuz ihtiyaçlarını tatmin etmek amacıyla ekonomik sistemler, politik sistemler, sosyal sistemler, endüstriyel sistemler, bilişim sistemleri ve daha bir çok yeni sistemler geliştirmiştir. Bu sistemleri geliştirirken, var olan doğal sistemleri analiz etmiş ve kendi sistemlerini yaratırken büyük ölçüde bu doğal sistemleri taklit etmiştir.

Çalışmamda ilk olarak sistem kavramı ele alınarak incelenmiştir. Sistem bir elemanlar kümesidir. Bu elemanlar, birbirleri ile ilişki ve etkileşim içinde belirli bir amacı gerçekleştirmek için faaliyette bulunurlar. Sistemlerin büyük bir çoğunluğu kompleks bir yapıya sahiptir. Bu yapıları nedeniyle incelenmeleri ve analiz edilmeleri bazen çok güç olmaktadır. İnsan, bu aşamada karşılaştığı güçlüğü aşmak için gerçek sistemi temsil eden bir model yaratma yoluna gitmiştir. Analiz etmek istediği sistemi, bu model üstünde incelemeler yaparak, anlamaya ve analiz etmeye çalışır. Çalışmamda sistem kavramını takiben, sistem modelleri konusu ele alınarak, modellerin kurulması ve sınıflandırılması konuları incelenmiştir. Sonraki aşamada ise, sistem analizi kısaca incelenmiştir. Sistem analizinin ayrıntılı olarak incelenmesi ise takip eden sistem yaklaşımı bölümünde, bir bütün içinde gerçekleştirilmiştir. Sistem yaklaşımı; aşamaları ve kullanılan araçlar itibariyle ele alınmıştır. Sistem yaklaşımında ve sistem analizinin uygulama aşamasında büyük önemi olan araçlar sistem akış diyagramları, program akış diyagramları ve karar tablolarıdır. Bu araçlar verilen örnekler ile canlandırılmış ve ayrıntılı olarak incelenmiştir.

Sistem yaklaşımının veya sistem analizinin, bir yaklaşım biçimi olarak sunulmasında veya bir ders olarak ilgili eğitim programlarında okutulmasında önemli bir sorun konunun soyut düzeyde kalmasıdır. Bu konudaki sıkıntıyı aşmak amacıyla gerçek sistemler ele alınıp, analizleri yapılabilir. Bu mümkün olmakla birlikte, aslında son derece güç ve bir hayli uzun bir çalışmayı gerektirir. Üstelik örnek olarak analizi yapılan sistem, kişilerin bilgi, görgü ve deneyimlerine uzak kalan bir konuda ise yapılan örnek çalışması faydadan çok, kişilerin olaya olan ilgisinin yok olmasına ve konudan uzaklaşmalarına neden olur. Bu nedenle,

çalışmamızda gerçek yaşamdan alınacak bu tür sistem analizi çalışmaları yerine, sistem yaklaşımı veya sistem analizinin temel araçlarının kullanılması konusunda, kapsamı daha sınırlı örnekler sunarak sistem analizi gibi soyut düzeyde kalabilecek bir konuyu daha somut bir ortamda canlandırmaya çalıştık. Bu amacı gerçekleştirirken sistem akış diyagramları ve program akış diyagramlarına konu olabilecek çeşitli örnekler ele alınarak bunların akış diyagramları çizilmiştir. Ayrıca karar tablolarına ait açıklamalar yine bu tarzdaki örnekler ile zenginleştirilmiştir.

İnsanoğlunun yarattığı uygarlığın en önemli unsurlarından bir tanesi de bilgisayardır. Bugün hemen, hemen tüm sorunlar bilgisayar yardımıyla veya doğrudan bilgisayar ortamında çözülmektedir. Çalışmamızda örnek olarak verilen sistem akış diyagramları, program akış diyagramları ve karar tablolarını takiben, bu örneklere ait C programlama dili kullanılarak yazılan bilgisayar programları sunulmuştur. Bu bilgisayar programları çalışmamızın konuya olan katkısını daha da belirginleştirirken, yazılan basit ama yararlı program örnekleri insanın hem sistem analizi, hem de C programlama dili konusundaki merakını uyandırıp, konunun kişilere daha yararlı olmasını sağladığı düşüncesindeyiz. Ele alınan olaylar, sistem analizinin araçları ile çözülürken, yazılan C programları ile bilgisayar ortamına aktarılan bu çözümler, olayın daha canlı ve anlamlı bir yapıya kavuşmasını sağlamaktadır. Son olarak sunulan "Dosyalama ve İstatistiksel Analiz Uygulaması" isimli bilgisayar programı ile C programlama dilinde bir dosyalama örneği sunulurken, aynı zamanda sistem yaklaşımı ve sistem analizinde yararlanılan istatistiksel yöntemlerden bazıları, farklı bir örnek olarak sunulmuştur.

Sonuç olarak, sistem analizi konusundaki yaklaşımımızın konuya ilgi duyan kişilere, kullanılan örnekler, çizilen akış diyagramları ve yazılan C programları aracılığı ile daha yararlı olacağı düşüncesini taşımaktayız.

Çankaya University Library

MARMARA ÜNİVERSİTESİ YAYINEVİ

YARARLANILAN KAYNAKLAR

- Borland, "Turbo C ++ Getting Started" , Borland International Inc., California, 1990.
- Borland, "Turbo C ++ User's Guide", Borland International Inc., California, 1990.
- Borland, "Turbo C ++ Programmer's Guide", Borland International Inc., California, 1990.
- Cutts, Geoff, "Structured Systems Analysis and Design Methodology", Paradigm Publishing Ltd., London, 1988.
- Davis, William S., "Management, Information, and Systems: An Introduction To Business Information Systems", West Publishing Company, Minneapolis, 1995.
- Ekinci, Fatih, "C ve Siz", Türkmen Kitabevi, İstanbul, 1996.
- Erkut, Haluk, "Sistem Analizi", Sistem Yayınları Dizisi 2, Kıyı Yayınları, İstanbul, 1989.
- Ertürk, Zeki, "Sistem Analizi", Ekonomist Yayınevi, Ankara, 1983.
- Esen, Öner, "İşletme Yönetiminde Sistem Yaklaşımı", İ.Ü. İşletme Fakültesi Yayın No:174, Bayrak Matbaacılık, İstanbul, 1985.
- Esen, Öner, "Programlamaya Giriş ve Basic", Baskan Yayınları, Başkent Ofset, İstanbul.
- İlyasoğlu, Eyüp, "Üretim Sistemlerinin Yönetimi", İ.İ.T.İ.A. Niht Sayr Yardım Vakfı Yayınları No: 365/598, İstanbul, 1982.
- Karagülle, İhsan, Pala, Zeytin, "Borland C ++ Builder", Türkmen Kitabevi, İstanbul, 1998.
- Kroeber, Donald W., Watson, Hugh j., "Computer-Based Information Systems : A Management Approach", Second Edition, Macmillan Publishing Company, New York, 1990.
- Schildt, Herbert, "C ++ The Complete Reference", Osborne McGraw-Hill, California, 1991.

Çankaya University Library

B015271
ÇANKAYA ÜNİVERSİTESİ KÜTÜPHANESİ

MARMARA ÜNİVERSİTESİ YAYINEVİ

Marmara Üniversitesi İletişim Fakültesi
Basın Yayın Teknikleri Ana Bilim Dalı
tarafından basılmıştır.

ISBN 975 – 400 – 177 - 4