

GAZİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

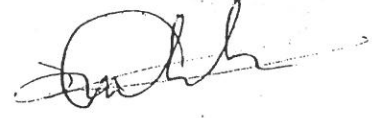
GENEL AMAÇLI BİR BULANIK MANTIK
DENETLEYİCİSİNİN TASARIMI

İSMAİL ATACAK

YÜKSEK LİSANS TEZİ
ELEKTRONİK-BİLGİSAYAR EĞİTİM ANABİLİM DALI

Ocak 1998
ANKARA

İsmail ATACAK tarafından hazırlanan GENEL AMAÇLI BİR
BULANIK MANTIK DENETLEYİCİSİNİN TASARIMI adlı bu tezin Yüksek
Lisans tezi olarak uygun olduğunu onaylarım.



Prof Dr. Doğan ÇALIKOĞLU

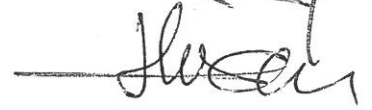
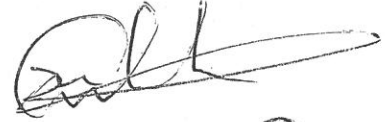
Tez Yöneticisi

Bu çalışma, jürimiz tarafından Elektronik Eğitimi Anabilim Dalında
yüksek lisans tezi olarak kabul edilmiştir.

Başkan...: Prof. Dr. Doğan ÇALIKOĞLU

Üye.....: Doç. Dr. Abdullah ÇAVUŞOĞLU

Üye.....: Yrd. Doç. Dr. H. Haldun GÖKTAŞ



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



GENEL AMAÇLI BİR BULANIK MANTIK DENETLEYİCİSİ
(Yüksek Lisans Tezi)

İsmail ATACAK

GAZİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
Ocak 1998

ÖZET

Bu çalışmada, on girişli bir çıkışlı genel amaçlı bir bulanık mantık denetleyicisi (GABDEN) tasarlanmış ve gerçekleştirilmiştir. GABDEN, veri sayısı bir ile on arasında değişen tüm sistemlerin denetlenmesinde kullanılabilir niteliktedir. Mantığın merkezini bir kişisel bilgisayar oluşturmaktadır. Bilgisayarın programı Quick Basic 7.1 dilinde hazırlanmıştır. Analog giriş ve çıkışlar için 16 girişli bir çıkışlı ve 12 bitlik bir hazır kart kullanılmıştır. GABDEN' nin uygulanmasına bir örnek olarak bir odanın sıcaklık denetimi verilmiştir. Giriş değeri olarak iç ortam sıcaklığı, dış ortam sıcaklığı, nem ve hava kalitesi verilmiştir. Çıkış değeri olarak ise ısıtıcı gücü gösterilmiştir.

Bilim Kodu : 225.10.01
Anahtar Kelimeler : Bulanık mantık, bulanık küme, uzman sistem
Sayfa Adedi : 85
Tez Yöneticisi : Prof. Dr. Doğan ÇALIKOĞLU

A GENERAL PURPOSE FUZZY LOGIC CONTROLLER
(M. Sc. Thesis)

İsmail ATACAK

GAZİ ÜNİVERSİTY
INSTITUTE OF SCIENCE AND TECHNOLOGY
JANUARY 1998

ABSTRACT

In this study, a general purpose ten input, one output fuzzy logic controller (called GABDEN) is designed and implemented. GABDEN can be used in the control of all systems which have upto ten inputs and one output. A personal computer constitutes the center of the logic. The program of the computer is prepared in the Quick Basic 7.1 language. A sixteen input one output and twelve bit ready made card is employed for analog input/output. The temperature control of a room is taken as an application example for GABDEN. In this application the inner temperature, the outer temperature, the humidity and the weather quality are chosen as input parameters and the required heater power is specified as the output.

Science code : 225.10.01

Key word : Fuzzy Logic, Fuzzy Set, Expert System

Page number : 85

Adviser : Prof. Dr. Doğan ÇALIKOĞLU

TEŞEKKÜR

Çalışmalarım boyunca yardımlarını ve desteğini esirgemeyen, engin bilgi ve fikirlerinden faydalandığım, danışmanım, çok değerli hocam Prof. Dr. Doğan ÇALIKOĞLU' na ve yardımları dokunan aşağıdaki çok değerli hocalarım ve arkadaşlarıma teşekkür ederim.

Yrd. Doç. Ömer Faruk BAY, Arş. Gör Vedat KIRAY, Teknik Eğitim Fakültesi Elektronik-Bilgisayar Eğitimi Bölümü Öğretim Elemanlarına.

İÇİNDEKİLER

	<u>Sayfa</u>
ÖZET	i
ABSTRACT	ii
TEŞEKKÜR	iii
İÇİNDEKİLER	iv
ŞEKİLLERİN LİSTESİ	viii
ÇİZELGELERİN LİSTESİ	x
SİMGELER VE KISALTMALAR	xi
1. GİRİŞ	1
2. BULANIK KÜMELERİN ESASLARI	5
2.1. Giriş	5
2.2. Bulanık Küme Teorisi	6
2.3. Bulanık Kümelerde İşlemler	9
2.3.1. Bileşim	9
2.3.2. Kesişim	11
2.3.3. Tümleneyen	14
2.3.4. Cebirsel Toplam	14
2.3.5. Cebirsel Çarpım	15
2.3.6. Üssel Kuvvet	15

2.3.7. İki Bulanık Kümenin Eşitliği	15
2.4. Klasik ve Bulanık İlişki	17
2.4.1. Max-Min Kompozisyonu	21
2.4.2. Max Çarpım Kompozisyonu	23
2.5. Bulanık Kümenin Türevleri	25
2.5.1. α - Kesişim	25
2.5.2. Destek	26
2.5.3. Normallik	26
2.5.4. Yükseklik	26
2.5.5. Genişletme	27
3. SÖZEL DEĞİŞKENLER	30
3.1. Giriş	30
3.2. Söz Dizimi	31
3.2.1. Yoğunluk	31
3.2.2. Genişlik	32
3.2.3. Zıt Yoğunluk	32
3.3. Sözel Niteleyiciler	32
4. ÜYELİK FONKSİYONLARI	39
4.1. Giriş	39
4.2. Üyelik Fonksiyonlarının Seçiminde Kullanılan Metotlar	39
4.2.1. Frekans Yaklaşımı	39

4.2.2. Uzaklık Yaklaşımı	40
4.2.3. Saaty' nin ikili Karşılaştırmalar Metodu ile Üyelik Fonksiyonlarının Hesaplanması	42
4.2.4. Olasılık Karakteristikleri ile Üyelik Fonksiyonlarının Hesaplanması.....	43
5. BULANIK MANTIK DENETLEYİCİLERİ.....	46
5.1. Giriş.....	46
5.2. Bulanık Mantık Denetleyicilerinin Genel Yapısı.....	46
5.2.1. Bulanıklaştırma Ünitesi.....	48
5.2.1.1. Üyelik Fonksiyonlarının Oluşturulması	49
5.2.2. Bilgi Tabanı.....	50
5.2.3. Çıkarım Ünitesi.....	52
5.2.4. Berraklaştırma Ünitesi.....	55
6. GABDEN' NİN TASARIMI VE GERÇEKLEŞTİRİLMESİ.....	61
6.1. Giriş.....	61
6.2. GABDEN' de Kontrol Basamaklarının Oluşturulması.....	62
6.2.1. Bulanıklaştırma İşlemleri	63
6.2.2. Kural Tabanının Oluşturulması	64
6.2.3. Çıkarım İşlemleri	65
6.2.4. Berraklaştırma İşlemleri	66
6.3. Bilgisayara Giriş-Çıkış Sistemi	67

6.3.1. 12 Bit A/D-D/A Giriş-Çıkış Kartı.....	67
6.3.1.1. A/D-D/A Giriş-Çıkış Kartının Özellikleri.....	67
6.3.1.2. Donanım Düzeni.....	68
6.3.1.3. Kısa Devre Bacaklarının Ayarları	70
6.3.1.4. Programlama ve Test Teknikleri.....	71
6.4. Uygulama Örneği.....	72
6.4.1. Bulanık Kontrol Basamaklarının Oluşturulması.....	72
6.4.1.1. Bulanıklaştırma İşlemleri	73
6.4.1.2. Kural Tabanının Oluşturulması	77
6.5. Deney.....	78
7. SONUÇ VE ÖNERİLER	84
KAYNAKLAR	
EKLER	
ÖZGEÇMİŞ	

ŞEKİLLERİN LİSTESİ

<u>Sekil No.</u>	<u>Açıklama</u>	<u>Sayfa</u>
2.1.	Klasik Kümelerin Bulanık Kümeler Haline Getirilmesi.....	7
3.1.	Hız Değişkenine Ait Hız Kümeleri.....	30
3.2.	Çok Niteleyicisinin Yorumlanması.....	35
5.1.	Bulanık Mantık Denetleyicisinin Blok Diyagramı	48
5.2.	Giriş Değerlerinin Bulanık Forma Dönüştürülmesi.....	49
5.3.	Kural Tabanı.....	52
5.4.	Minimum İlişki Yöntemi İle Kuralların Değerlendirilmesi.....	55
5.5	Üyelik Fonksiyonlarının Max Noktaları İle Berraklaştırma	56
5.6.	Merkez Yöntemi İle Berraklaştırma İşlemi	56
5.7.	Ağırlıklı Ortalama Yöntemi ile Berraklaştırma İşlemi	57
5.8	Max Noktaların Ortalaması Yöntemi ile Berraklaştırma İşlemi	58
5.9.	Geniş Alan Merkezi Metodu İle Berraklaştırma İşlemi.....	58
5.10	İlk veya Son Yükselti Metodu İle Berraklaştırma İşlemi.....	59
6.1.	Bulanık Mantık Denetleyicisinin Sistem Bağlantısı.....	62
6.2	A/D-D/A Karta Ait 25 Bacaklı Konnektörün Bacak Dizilişi.....	69
6.3	A/D-D/A Kartın Yerleşim Düzeni	70
6.4	İç Sıcaklık Bilgisine Ait Üyelik Fonksiyonları.....	73
6.5	Dış Sıcaklık Bilgisine Ait Üyelik Fonksiyonları	74

6.6.	Nem Bilgisine Ait Üyelik Fonksiyonu.....	76
6.7.	Hava Kalite Bilgisine Ait Üyelik Fonksiyonu.....	76
6.8.	Çıkışa Ait Üyelik Fonksiyonları.....	77
6.9	Denetleyici Sisteme Ait Programın Akış Diyagramı	83

ÇİZELGELERİN LİSTESİ

<u>Çizelge No.</u>	<u>Açıklama</u>	<u>Sayfa</u>
6.1.	Denetleyicide Kural Tabanının İkili Matrisler Halinde Sıralanması	64
6.2	Üç Girişli Bir Sistem İçin Oluşturulan Kural Tabanı	65
6.3	A/D-D/A Kartın Belirleyici Özellikleri	68
6.4	A/D-D/A Kartın Adreslerinin Ayrıntılı Karşılıkları	68
6.5	D Tipi Konnektör Bacak Sırası ve Fonksiyonu.....	69
6.6.	İç Ortam Sıcaklığına Ait Üyelik Fonksiyonlarının Oluşturulması.....	74
6.7.	Dış Ortam Sıcaklığına Ait Üyelik Fonksiyonlarının Oluşturulması.....	75
6.8.	Kural Tabanının Oluşturulması.....	77

SİMGELER VE KISALTMALAR

<u>Simge</u>	<u>Açıklama</u>
$\subseteq$	Kapsar
$\cap$	Kesişim
$\cup$	Bileşim
$\neg$	Değil
ε	Elemanıdır
Σ	Karetezyen Toplam
μ	Üyelik Fonksiyonu
$\oplus$	Bulanık Toplam
$\emptyset$	Boş Kümesi
u^*	Berraklaştırılmış Değer
W_i	Üyelik Ağırlığı
R	Bulanık Toplam
$\leq$	Küçük Eşittir

<u>Kısaltmalar</u>	<u>Açıklama</u>
RAM	Rastgele Erişimli Bellek (Random Access Memory)
ROM	Yalnızca Okunan Bellek (Read Only Memory)
I/O	Giriş-Çıkış (Input / Output)

A/D	Analog Dijital Dönüştürücü
D/A	Dijital Analog Dönüştürücü

1.GİRİŞ

Bu çalışmada, on girişli bir çıkışlı genel amaçlı bir bulanık mantık denetleyicisi tasarlanmış ve gerçekleştirilmiştir. Bu denetleyici, GABDEN (Genel Amaçlı Bulanık Denetleyici) olarak isimlendirilmiştir. GABDEN, veri sayısı bir ile on arasında değişen tüm sistemlerin denetlenmesinde kullanılabilecek niteliktedir. GABDEN ile deneysel amaçlı yapılacak çalışmalarda, bulanık mantık denetleyicilerinin klasik mantık denetleyicilerine göre üstünlükleri de gözlenebilecektir. Bu şekilde özellikle laboratuvar uygulamaları için esnek bir sistem kazanılmıştır.

Bulanık mantık, insanların esnek ve değişken yapısını dikkate alan bir kontrol algoritmasıdır. Bilgiler arasında sebep-sonuç ilişkisi kurarak doğru ve mantıksal bir sonuç üretir. Bu işlemin yapılabilmesi için ilk olarak sisteme ait verilerin belirlenmesi gerekmektedir. Bu veriler belirli sınırlar içerisinde gruplandırılarak bulanık kümeler haline getirilir. Bulanık mantıkta bu gruptama işlemi üyelik fonksiyonları vasıtası ile gerçekleştirilir. Üyelik fonksiyonları verilerin ne oranda hangi kümeye ait olduğunu tespit eden bir nitelidir. Daha sonra denetlenecek sistemle ilgili tüm olası durumlar dikkate alınarak kural tabanı oluşturulur. Bu kurallar bir kontrol algoritması ile değerlendirilerek sisteme gönderilecek bilgi tespit edilir.

Bulanık mantık denetleyicilerinin ilk önemli uygulaması Londra Üniversitesinden Prof. Dr. Mamdani tarafından buhar türbününün denetlenmesi üzerinde yapılmıştır. Prof. Dr. Mamdani, bulanık mantıkla yaptığı bu çalışma ile türbünün basıncı ve motor hızının başarılı bir şekilde denetlenebileceğini göstermiştir. Bu ilk uygulamadan sonra bulanık kontrole olan ilgi artmış ve değişik alanlarda kullanılmaya başlamıştır. King ve Mamdani, bulanık mantık kullanarak kimyasal bir reaktörün sıcaklığını kontrol etmişlerdir. Shigeyasu ve

Li, servo sistemler için bulanık denetleyicilerin tasarımını gerçekleştirerek bulanık denetleyicilerin PID denetleyicilerden daha iyi sonuç verdiğini göstermişlerdir. Rahmani, bina ısıtma sistemleri için bulanık denetleyicileri kullanarak oldukça başarılı sonuçlar almıştır. (Baba, 1995; Kıray, 1997)

Bundan sonraki en önemli gelişme çimento sanayisinde olmuştur. Bir Danimarka firması, değirmenin oksijen ve sıcaklık oranlarını kontrol eden bir bulanık denetleyici geliştirmiştir. 1987 yılında Hitachi firması, Sedai Metro sisteminde çalışan trenleri bulanık mantıkla kontrol ederek başarılı sonuçlar elde etmiştir. (Kaynak, 1992)

Bulanık mantık teorisi, bugüne kadar bir çok bilim dalında uygulanmıştır. Yapay zeka, uzman sistemler, kontrol teorisi, kalite kontrol, ürün planlaması ve seçimi, optimum sistem planlaması, taşıma-ulaşım, network, oyunlar kuramı, bankacılık-finance, ziraat bunlardan bazılarıdır.

GABDEN ile herhangi bir sistemin denetlenebilmesi için o sisteme ait bilgilerin tanımlanması gerekmektedir. Denetlenecek sisteme ait bilgilerin tanımlanabilmesi ve bu sistemin denetlenebilmesi için iki ayrı menü hazırlanmıştır. İlk menüde yardım, yeni sistem, eski sistem ve çıkış olmak üzere dört ayrı seçenek bulunmaktadır. Yardım seçeneğinde denetleyicinin kullanılmasına ait bilgiler yer almaktadır. Yeni sistem seçeneği seçildiği zaman, önceden tanımlanmış sisteme ait tüm bilgiler hafızaya kaydedilerek ikinci menüye geçilir. Eski sistem seçeneği seçildiğinde daha önce tanımlanan sisteme ait bilgiler dikkate alınarak sistem denetlenmeye başlar. Sonuç hatanın zamana göre değişimini gösteren bir grafikte gösterilir. İkinci menü tanım1, tanım2 ve yürütme olmak üzere üç ayrı seçenektir oluşmaktadır. Bu menüde tanım1 seçeneğinde; ilgili sisteme ait üyelik fonksiyonları, üyelik kümeleri ve üyelik fonksiyonlarına ait sabitler tanımlanmaktadır. Tanım2 seçeneğinde giriş

ve çıkış için tanımlanmış üyelik kümesine ait alt sınır, üst sınır ve kural tabanı değerleri tanımlanmaktadır. Denetlenecek sisteme ait üyelik fonksiyonları, en fazla üç ayrı aralıkta ifade edilebilir. Üyelik fonksiyonları iki veya daha az bir aralık için tanımlanacaksa diğer aralık değerleri girilmemelidir. Fonksiyon tanımlanırken; değişken olarak "x", sabitler olarak a, b, c gibi harfler seçilmelidir. Üyelik kümelerinin sayısı, bir ile yedi arasında değişen sözel değişkenler tanımlanacak şekilde düzenlenmiştir. Denetlenecek bir sistemin, hassasiyetini artırmak için üyelik kümesi sayısı mümkün olduğu kadar fazla tutulmalıdır. Üyelik fonksiyonunda a, b, c gibi harflerle isimlendirilen sabit değerler, sabitler seçeneği seçilerek tanımlanmaktadır. Böylece her bir üyelik kümesi için ayrı ayrı üyelik fonksiyonu oluşturulur. Aralık seçeneğinde üyelik kümelerine ait alt ve üst sınır değerleri tanımlanmaktadır. Kural tabanı seçeneğinde ise denetlenecek sistemi sonuca ulaştıran kurallar tanımlanmaktadır. PC' nin algılayıcıdan aldığı sinyallerle irtibatını, Decision-Computer International Co. Ltd. firmasının üretimi olan arabirim devresi sağlamaktadır.

Örnek bir uygulama olarak bir odanın sıcaklık denetlemesi seçilmiştir. Denetleyicinin giriş değişkenleri olarak iç ortam sıcaklığı, dış ortam sıcaklığı, nem ve hava kalitesi seçilmiştir. Çıkış değişkeni olarak da ısıtıcı gücü seçilmiştir. Gönderilen sinyal triyak' ın geytine uygulanarak ortamın ısısı 19 derecede sabit tutulmuştur. Sistemin giriş ve çıkış parametrelerine ait 7 ayrı üyelik kümesi tanımlanmıştır. Çıkış sabit fonksiyon olarak tanımlanmıştır. Bunun sebebi, diğer fonksiyonlarda ağırlık merkezinin hesaplanmasının zor oluşundan kaynaklanmaktadır. Denetleyici sistemin çıkarım işlemlerinde max-min metodu, berraklaştırma işlemlerinde ise ağırlık merkezi metodu kullanılmıştır.

GABDEN ile yapılan denetim işlemlerinin klasik mantık denetleyicilerine göre daha iyi bir sonuç verdiği gözlenmiştir. Tasarlanan algoritma değişik

sistemlerin denetlenmesini sağlayacak şekilde yazıldığından, deneysel amaçlı olarak da kullanılabilir.

Bölüm-2' de bulanık kümelerin esasları üzerinde durularak bulanık kümelerle ilgili alt kavramlar anlatılmıştır.

Bölüm-3' de bulanık kümelerde önemli bir yere sahip olan sözel değişkenler ve bunun alt kavramları anlatılmıştır.

Bölüm-4' de üyelik fonksiyonları ve bunun hesaplanmasında kullanılan metotlar anlatılmıştır.

Bölüm-5' de bulanık mantık denetleyicileri ve bunu oluşturan üniteler kapsamlı bir şekilde anlatılmıştır.

Bölüm-6' da GABDEN' nin tasarlanması ve gerçekleştirilmesi ile GABDEN' le yapılan bir uygulama örneği anlatılmıştır.

2. BULANIK KÜMELERİN ESASLARI

2.1. Giriş

Bulanık küme teorisi, ilk kez 1965 yılında Lotfi Asker Zadeh tarafından belirsizliklerin matematiksel bir model olarak ifade edildiği teori olarak ortaya çıkmıştır. Prof. Dr. Lotfi A. Zadeh' in 1965 yılında yaptığı bu çalışma, bundan sonraki yıllarda bu alanda çalışan bilim adamları için altın bir kaynak olmuştur.

Günümüzde en çok karşılaştığımız problemlerden biri, yeterli ve gerekli bilgi elde edemediğimiz konularda karar vermek ve tahminde bulunmaktır. Yeterli bilgi elde edilememesinin en büyük sebeplerinden biri konunun karmaşık bir yapıya sahip olmasından kaynaklanmaktadır. Bu tür karmaşıklıkların çözümü; verilen bilgi ve belirsizlik miktarı arasındaki tatmin edici uzlaşmayı sağlayarak, karmaşıklığı basitleştirmekten geçmektedir. Bir başka seçenek ise verilebilir belirsizlik miktarını artırmaktır. Örneğin hızla ilgili bir bilgi verirken, hız miktarını tam olarak belirtmek yerine belirsizlik değeri daha yüksek olan "hızlı" yada "yavaş" tanımlamasını yapmak daha uygun olacaktır. Bu şekilde sınırları tam olarak bilinmeyen bir veri hakkında da yorumlama yapmamız mümkün olacaktır. (Bağcı oğulları, 1995)

İnsan beyni sayısal bir işlemi bir kaç dakikada yapmasına rağmen, idrak etmeye yönelik olayları bir kaç saniyede çözebilecek kadar hızlıdır. Örneğin yolda giden bir şoför aniden fren yapabilir. Şoför o an yolun kayganlığı, hız gibi etmenleri göz önüne alarak, geçmişte kazandığı deneyimlerden tehlikeyi idrak ettiğinden dolayı, tehlikeye anında karşı koyar. Bilgisayarlar ise çok karmaşık sayısal işlemleri bir kaç saniyede yapabilmelerine ramen, idrak etmeye ve deneyimlere yönelik işlemlerde yetersiz ve oldukça yavaşlardır. Bu

tür bir sakıncaları ortadan kaldırmak için Zadeh tarafından bulanık küme teorisi ortaya atılmıştır. Bulanık küme teorisi ile insan oğlunun bu özellikleri kullanılarak karmaşık problemlerin formülasyonu ve çözümü daha kolay ve sade bir şekilde elde edilmiştir. (Başbuğ, 1997)

Bulanık kümeler klasik kümelerin özelleştirilmiş bir halidir. Klasik kümelerde olduğu gibi sadece 0,1 değerleri kullanılmayıp, bunun arasındaki ara değerleride kullanarak işlemler gerçekleştirilmektedir. Bunun dışında klasik kümelerde kullanılan tüm özellikler bulanık kümeler içinde kullanılabilir.

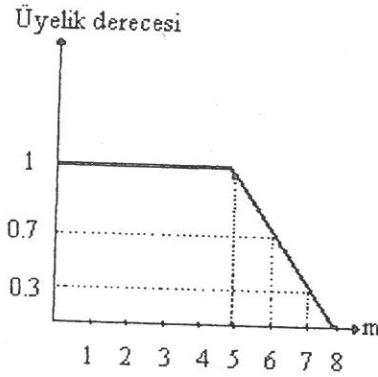
2.2. Bulanık Küme Teorisi

Günlük hayatımızda kesin olmayan ifadelerin matematiksel olarak modellenmesini sağlayan teoreme “Bulanık Küme Teorisi” denir. Buradaki kesin olmayan ifadelerden kastımızı bir örnekle açıklayalım. Uzak - yakın kesin sınırları ile belirtilen iki ifadenin arasındaki az uzak, az yakın gibi ifadeler belirsizliğe bir örnek olarak gösterilebilir.

Bulanık mantık temel’ olarak insan düşünüş şeklini örnek almaktadır. Dolayısıyla bu mantığın küme elemanları, insan mantığına çok yakın yada aynı olan elemanlardır. Burada yine geleneksel mantıkta olduğu gibi “0” ve “1” değerleri vardır. Ancak bulanık mantık yalnızca bu değerlerle yetinmeyip, bunların ara değerleride kullanarak örneğin bir uzaklığın yalnızca uzak yada yakın olduğunu belirtmekle kalmayıp ne kadar uzak yada yakın olduğunda söyler. Bulanık mantıkta olaylar klasik mantığın tersine daha esnektir. Bu haliyle bulanık mantık, soğuk- sıcak, hızlı - yavaş, yüksek- alçak gibi esnek niteleyicilerle yumuşatarak gerçek dünyaya uyarlar. Bulanık küme teorisini daha iyi anlayabilmek için aşağıda vereceğimiz örneği inceleyelim. (Bağcıoğulları, 1996)

Örnek 2.1:

Ahmet' in herhangi bir marka arabası olduğunu farz edelim. Güvenlikle taşıyacağı insan sayıları 1, 2, 3, 4, 5 (doğal sayılar) A kümesinin elemanlarıdır. Ama bu sayılara 6, 7 gibi sayılarda eklenebilir. Bu iki sayının A kümesine aitlik derecesi 1' in altında olacaktır. 8, 9, 10 gibi sayılar kesinlikle bu kümeye ait olamaz. 6, 7 rakamlarından 6' nın A kümesine aitlik derecesi 7' ye göre daha fazladır. 6' nın A' ya aitlik derecesi $\mu_{A(6)} = 0,7$ iken 7' nin aitlik derecesi $\mu_{A(7)} = 0,3$ ' dir. Bunu grafik olarak şekil 2.1' deki gibi ifade ederiz.



Şekil 2.1. Klasik kümelerin bulanık küme haline getirilmesi

Karakteristik (üyelik) fonksiyonunun değer kümesi $[0-1]$ aralığında sürekli her gerçek sayıyı alabilecek şekilde tanımlanmış ise, bu kümeye “Bulanık Küme” denir. Bulanık kümeler üyelik fonksiyonları ile temsil edilirler. X klasik evrensel kümeyi, x ise bu kümeye ait genel bir elemanı göstermektedir. A gibi bir alt kümenin oluşturabilmesi için, tanım kümesinde verilen değerlere bağlı olarak değer kümesinde $[0-1]$ aralığında her gerçek sayı sağlayan karakteristik fonksiyona “üyelik Fonksiyonu” denir. A gibi bir kümenin üyelik fonksiyonu μ_A ile gösterilir. Tanım kümesindeki değerlere bağlı olarak, değer kümesinde elde edilen değerlere “üyelik derecesi” adı verilir. Bu, A

kümesine ait bir x elemanı için $\mu_A(x)$ ile gösterilir. A bulanık kümesine ait bir x elemanın üyelik derecesine göre yorumlanması aşağıdaki gibi yapılmaktadır.

1) x elemanının A kümesine ait üyelik derecesi "1" ($\mu_A(x)=1$) ise x , A kümesinin mutlak elemanıdır.

2) x elemanının A kümesine ait üyelik derecesi "0" ($\mu_A(x)=0$) ise x , A kümesinin elemanı değildir.

3) x elemanının A kümesine üyelik derecesi $[0 - 1]$ aralığındaki değerlerden herhangi birini almış ise, şu iki durum incelenir.

a) $\mu_A(x)$ değeri 1'e yakın bir değer ise, x elemanının A kümesine üyeliği daha güçlüdür.

b) $\mu_A(x)$ değeri 0'a yakın bir değer ise, x elemanının A kümesine üyeliği zayıftır. Üyelik derecesi, bulanık küme için tanımlanmış üyelik fonksiyonunun tipine göre değişecektir.

X evrensel kümeyi, x bu kümeye ait genel bir eleman ve $\mu_A(x)$ ise A kümesine ait x elemanının üyelik derecesi gösterebilir. Bu duruma göre A bulanık kümesi;

$$A = \{(X, \mu_A(x) \mid x \in X\} \quad (2.1)$$

biçiminde gösterilir. Örnek 1'e "Eş. 2.1" uygulanırsa;

$$N = \{(1, 1), (2, 1), (3, 1), (4, 1), (5, 1), (6, 0.7), (7, 0.3)\}$$

bulanık kümesi elde edilir. Bu sıralı ikililerin ikinci terimleri, birinci terimlerin N kümesine ait olan üyelik derecelerini göstermektedir. Bulanık kümeler klasik kümelerin bir uzantısı olduğundan klasik kümeler için yapılan tanımlar bulanık kümeler içinde yapılabilir. (Zadeh, 1987)

2.3. Bulanık Kümelerde İşlemler

2.3.1. Bileşim

X evreninde A ve B gibi iki bulanık alt küme tanımlansın. Bu bulanık alt kümelerin üyelik dereceleri $\mu_A(x)$ ve $\mu_B(x)$ olarak verilsin. A ve B' nin Bileşimi bir fonksiyon olarak şu şekilde gösterilmektedir.

$$U: [0,1] \times [0,1] \rightarrow [0,1] \quad (2.2)$$

Evrensel kümedeki her bir x değeri için A ve B bulanık kümelerinin Bileşimi "Eş. 2.3" de olduğu gibi gösterilmiştir.

$$\mu_{A \cup B}(x) = U[\mu_A(x), \mu_B(x)] \quad (2.3)$$

Bileşim işlemi aşağıda belirtilen aksiyomlar dikkate alınarak gerçekleştirilmektedir.

Aksiyom U1: Bileşim işlemi sınır şartlarında, klasik kümelerde olduğu gibi gerçekleştirilmektedir.

$$U(0,0) = 0; U(0,1) = U(1,0) = U(1,1) = 1 \quad (2.4)$$

Aksiyom U2: Bileşim işleminde değişme (commutative) özelliği vardır.

$$U(a,b) = U(b,a) \quad (2.5)$$

Aksiyom U3: bileşim işleminde aşağıdaki eşitliği sağlayan işlem monotoniktir.

$$a \leq \neg a \text{ ve } \neg b \leq b \text{ ise } U(a,b) \leq U(\neg a, \neg b) \quad (2.6)$$

Aksiyom U4: Bileşim işleminde çağrışımılık (associative) özelliği vardır.

$$U(U(a,b), c) = U(a, U(b,c)) \quad (2.7)$$

Aksiyom U5: U sürekli bir fonksiyondur.

Aksiyom U6: Birbirinin aynısı iki kümenin bileşimi yine kendisine eşittir.

$$U(a,a) = a \quad (2.8)$$

“Eş. 2.8” yarger sınıfı ile tanımlanırsa ;

$$U_w(a,b) = \min [1, (a^w + b^w)^{1/w}] \quad (2.9)$$

eşitliği elde edilir. Burada w , $0 - \infty$ aralığında bir değer alır. Bileşim işlemini bulmak için; w sonsuza giderken limit işlemi uygulanmaktadır.

$$\lim_{w \rightarrow \infty} \min [1, (a^w + b^w)^{1/w}] \quad (2.10)$$

“Eş. 2.10” çözülerek aşağıdaki sonuç elde edilmiştir.

$$\mu_{A \cup B}(x) = \max (\mu_A(x), \mu_B(x)) \quad (2.11)$$

Kontrol teorisinde bileşim işleminin yerine lojik “ VEYA ” operatörü kullanılmaktadır.

Örnek 2.2:

X evreninde, A ve B bulanık kümeleri aşağıdaki gibi tanımlanmıştır.

$$A = \{1/2 + 0.5/3 + 0.3/4 + 0.2/5\}$$

$$B = \{0.5/2 + 0.7/3 + 0.2/4 + 0.4/5\}$$

$A \cup B$ kümesi, A ve B kümelerinde aynı isim altında bulunan elemanların maksimumu alınarak elde edilmiştir.

$$A \cup B = \{1/2 + 0.7/3 + 0.3/4 + 0.4/5\}$$

2.3.2 Kesişim

Bulanık kesişim, bulanık bileşimle paralellik gösterir. X evreninde tanımlanmış A ve B gibi iki bulanık alt kümenin kesişimi, “ $(A \cap B)$ ” şeklinde gösterilir. Kontrolde “VE” bağlacına karşılık gelir. İki kümenin kesişimi bir fonksiyon olarak şu şekilde gösterilmektedir.

$$\dot{I}: [0,1] \times [0,1] \rightarrow [0,1] \quad (2.12)$$

Belirtilen bu fonksiyon, A ve B bulanık kümelerinden seçilen x elemanının üyelik derecesi ile ilgilidir. Fonksiyon, $A \cap B$ kümesindeki elemanların üyelik

derecelerini tanımlar. Böylece $A \cap B$ ' nin üyelik dereceleri cinsinden gösterimi;

$$\mu_{A \cap B}(x) = I(\mu_A(x), \mu_B(x)) \quad (2.13)$$

şeklinde elde edilecektir. Bir bulanık kesişimi gerçekleştirirken, aşağıdaki aksiyomlar dikkate alınır.

Aksiyom İ.1: Kesişim işlemi sınır şartlarında klasik kümelerde olduğu gibi gerçekleştirilir.

$$I(1,1) = 1; I(1,0) = I(0,1) = I(0,0) = 0 \quad (2.14)$$

Aksiyom İ.2: Kesişim işleminde değişme özelliği vardır.

$$I(a,b) = I(b,a) \quad (2.15)$$

Aksiyom İ.3: Kesişim işlemi monotoniktir.

$$a \leq \neg a \text{ ve } b \leq \neg b \text{ ise } I(a,b) \leq I(\neg a, \neg b) \quad (2.16)$$

Aksiyom İ.4: Kesişim işleminde çağrışımlılık (associative) özelliği vardır

$$I(I(a,b),c) = I(a, I(b,c)) \quad (2.17)$$

Aksiyom İ.5: Kesişim işlemi, sürekli bir fonksiyondur.

Aksiyom İ.6: Birbirinin aynısı iki kümenin kesişimi yine kendisine eşittir.

$$\dot{I}(a, a) = a \quad (2.18)$$

Kesişim işlemi yager sınıfı ile aşağıdaki gibi tanımlanmaktadır.

$$\dot{I}_w(a, b) = 1 - \min [1, ((1-a)^w + (1-b)^w)^{1/w}] \quad (2.19)$$

Kesişim işlemi bulmak için “Eş. 2.19 ” un w sonsuza giderken limiti alınmıştır.

$$\lim_{w \rightarrow \infty} \dot{I}_w = \lim_{w \rightarrow \infty} (1 - \min [1, ((1-a)^w + (1-b)^w)^{1/w}]) \quad (2.20)$$

$a \leq b$ ise $1-a \geq 1-b$ ve

$$\dot{I}_\infty(a, b) = 1 - [1-a] = a$$

$a \geq b$ ise $1-a \leq 1-b$

$$\dot{I}_\infty(a, b) = 1 - [1-b] = b$$

$$\dot{I}(a, b) = \min (a, b) \quad (2.21)$$

“ Eş. 2.21 ” A ve B kümeleri için düzenlenirse;

$$\mu_{A \cap B}(x) = \min (\mu_A(x), \mu_B(x)) \quad (2.22)$$

şeklinde elde edilmiştir. (Klir and Folger, 1988)

Örnek 2.2’ deki değerler için kesişim işlemi uygulanırsa, $A \cap B$ kümesi aşağıdaki gibi elde edilecektir.

$$A \cap B = \{ 0.5/2 + 0.5/3 + 0.2/4 + 0.2/5 \}$$

2.3.3. Tümleyen (Değilleme)

X evrensel kümesinde tanımlanmış A bulanık kümesinin tümleyeni, $\neg A$ ile gösterilir. Kontrolde, “ değil ” (NOT) işlemine karşıl gelir. Üyelik fonksiyonu olarak, aşağıdaki eşitlikte olduğu gibi ifade edilebilir.

$$\mu_{\neg A}(x) = 1 - \mu_A(x) \quad x \in X \quad (2.23)$$

Örnek 2.3:

X evrensel kümesi ve A bulanık alt kümesi aşağıdaki gibi verilmiştir.

$$X = \{1, 2, 3, 4, 5\}$$

$$A = \{0/1 + 0.5/2 + 0.3/3 + 0.2/4 + 0.4/5\}$$

A bulanık kümesinin tümleyeni, A kümesindeki elemanların 1’ den çıkarılması sonucunda elde edilmiştir.

$$\neg A = \{ 1/1 + 0.5/2 + 0.7/3 + 0.8/4 + 0.6/5 \}$$

2.3.4. Cebirsel Toplam

X evreninde tanımlanmış A ve B gibi iki bulanık kümenin cebirsel toplamı $(A \oplus B)$ olarak gösterilir. Üyelik fonksiyonu olarak hesaplanması “Eş. 2.24” te verilmiştir.

$$\mu_{A \oplus B}(x) = \mu_A(x) + \mu_B(x) - \mu_A(x) \cdot \mu_B(x) \quad x \in X \quad (2.24)$$

2.3.5. Cebirsel çarpım

X evreninde tanımlanmış A ve B bulanık kümelerinin cebirsel çarpımı μ_{AB} ile gösterilip, aşağıdaki gibi hesaplanmaktadır.

$$\mu_{AB}(x) = \mu_A(x) \cdot \mu_B(x) \quad x \in X \quad (2.25)$$

2.3.6. Üssel kuvvet

A kümesi x evreninde tanımlanmış bir bulanık kümeyi, "b" ise herhangi bir pozitif tam sayıyı ifade etsin. Bu durumda A kümesinin b'ninci kuvveti A^b olarak gösterilip, üyelik fonksiyonu aşağıdaki gibi hesaplanmaktadır.

$$\mu_{A^b}(x) = [\mu_A(x)]^b \quad x \in X, b \in \mathbb{N}^+ \quad (2.26)$$

2.3.7. İki bulanık kümenin eşitliği

X evreninde, A ve B gibi iki bulanık küme tanımlansın. A bulanık kümesinin herbir elemanı B bulanık kümesinin elemanlarına eşitse, bu iki küme eşittir denir. Üyelik fonksiyonu olarak aşağıdaki gibi ifade edilmektedir.

$$\mu_A(x) = \mu_B(x) \quad x \in X \quad (2.27)$$

Klasik küme teorisinde gösterilen bir çok cebirsel özellik, bulanık kümeler içinde gösterilebilir. Bunlar, maddeler halinde şu şekilde sıralanabilir. (Kichert, 1978)

1) Kesişim ve bileşim işleminde değişim özelliği

$$A \cap B = B \cap A$$

$$A \cup B = B \cup A$$

2) Kesişim ve bileşim işleminde çağrışımlılık özelliği

$$A \cap (B \cap C) = (A \cap B) \cap C$$

$$A \cup (B \cup C) = (A \cup B) \cup C$$

3) Kesişim ve bileşim işleminde aynılık (idempotent) özelliği

$$A \cap A = A$$

$$A \cup A = A$$

4) Kesişim ve bileşim işleminde dağılma özellikleri

a) Kesişim işleminin bileşim işlemi üzerinde dağılma özelliği

$$A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$$

b) Birleşim işleminin kesişim işlemi üzerinde dağılma özelliği

$$A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$$

5) De Morgan teoremi

$$\neg(A) = \neg A$$

$$\neg(A \cap B) = \neg A \cup \neg B$$

$$\neg(A \cup B) = \neg A \cap \neg B$$

$$6) \emptyset : \mu_{\emptyset(x)} = 0 \quad \forall x \in X$$

$$7) E : \mu_{E(x)} = 1 \quad \forall x \in X$$

$$8) A \cup \emptyset = A$$

$$9) A \cap \emptyset = \emptyset$$

$$10) A \cup X = A$$

$$11) A \cap X = A$$

$$12) \text{Eğer } A \subseteq B \subseteq C \text{ ise } A \subseteq C \text{ olur.}$$

X evrensel kümesinde tanımlanmış A bulanık kümesinin kendisi ile evriğinin bileşimi “Evrensel Küme”, kesişimi ise “Boş Küme” değildir.

$$A \cup \neg B \neq X \quad \text{ve} \quad A \cap \neg B \neq \emptyset$$

2.4. Klasik ve Bulanık İlişki

Bir klasik ilişki, iki veya daha fazla kümenin elemanları arasındaki etkileşimin kuvvetini göstermek olarak genelleştirilebilir. Klasik küme teorisinde bu etkileşimin kuvveti 1 veya 0' dır.

X ve Y iki klasik kümeyi göstermek üzere; bu kümelerin kartezyen çarpımı $X \times Y$ kümesi ile gösterilip, her bir eleman sıralı ikili şeklinde ola-

caktır. Birinci eleman X kümesinden, ikinci eleman Y kümesinden seçilir. Formül olarak şu şekilde gösterilmiştir.

$$X \times Y = \{ (x,y) \mid x \in X \text{ ve } y \in Y \} \quad (2.28)$$

Eğer $X \neq Y$ ise $X \times Y \neq Y \times X$ olur.

$\{ X_i \mid i \in N_n \}$ şeklinde verilen bir klasik küme topluluğu için kartezyen çarpım $X_1 \times X_2 \times X_3 \dots X_n$ şeklinde genelleştirilebilir.

$X_1, X_2, X_3, \dots, X_n$ klasik kümeleri arasında tanımlanmış ilişki kartezyen çarpımın bir alt kümesidir. Bu ilişki $R(X_1, X_2, \dots, X_n)$ veya kısaltılarak $R(X_i \mid i \in N)$ formülüyle gösterilir.

$$R(X_1, X_2, \dots, X_n) \subset X_1 \times X_2 \times \dots \times X_n \quad (2.29)$$

Burada $X, X_2, \dots, X_n$ klasik kümeleri arasındaki ilişki için, $X_1 \times X_2 \times \dots \times X_n$ kartezyen çarpımı bir evrensel küme olarak gösterilmiştir. Kümeler arasında gösterilen bu ilişki alt küme, bileşim, kesişim, tersleme gibi temel kavramlar kullanılarak elde edilir. Kümeler arasında gösterilen R klasik ilişkisi bir fonksiyon olarak aşağıdaki gibi tanımlanmıştır.

$$\mu R(X_1, X_2, \dots, X_n) = \begin{cases} 1 \rightarrow \text{Eğer}(X_1, X_2, \dots, X_n) \in Rise \\ 0 \rightarrow \text{Eğer}(X_1, X_2, \dots, X_n) \notin Rise \end{cases} \quad (2.30)$$

Bunu bir örnek ile açıklayalım. R ilişkisi, tüm kadınların kümesi ile tüm erkeklerin kümesi arasındaki bulanık ilişkiyi gösterebilir. Kadınların oluşturduğu kümenin elemanları ile erkeklerin oluşturduğu kümenin elemanları arasında oluşturulan her sıralı ikili, R bulanık ilişkisinin bir elemanı olduğu için "1" değerini alır. Eğer bu ilişki 3, 4 veya 5 küme arasında gerçekleştirili-

yor ise; elde edilen ilişkiler sırasıyla 3' lü 4' lü veya 5' li elemanları ihtiva eder. Genel olarak n küme arasında tanımlanan bir ilişki n boyutlu olarak isimlendirilmektedir. n boyutlu bir ilişki $R (X_1, X_2, \dots, X_n)$ şeklinde ifade edilip, üyelik fonksiyonu olarak $\mu_R = (\mu_{R1}, \mu_{R2}, \dots, \mu_{Rn})$ şeklinde gösterilmektedir.

Örnek 2.4:

R ; X, Y, Z kümeleri arasındaki ilişkiyi göstermek üzere;

$$X = \{ \text{English, French} \}, Y = \{ \text{Dollar, Pound, Franck, Mark} \}$$

$$Z = \{ \text{United States, France, Kanada, Britain, Germany} \}$$

olarak tanımlanmaktadır. Bu verilerden üç küme arasındaki ilişki;

$$R(X, Y, Z) = \{ (\text{English, Dollar, United States}); (\text{French, Franck, France}); (\text{English, Dollar, Canada}); (\text{France, Dollar, Kanada}); (\text{English, Pound, Britain}) \} \text{ olarak elde edilir.}$$

	US	FR	CN	BRT	GR
<i>dollar</i>	1	0	1	0	0
<i>pound</i>	0	0	0	1	0
<i>frank</i>	0	0	0	0	0
<i>mark</i>	0	0	0	0	0

English

	US	FR	CN	BRT	GR
<i>dollar</i>	0	0	1	0	0
<i>Pound</i>	0	0	0	0	0
<i>Frank</i>	0	1	0	0	0
<i>Mark</i>	0	0	0	0	0

French

Birden fazla bulanık küme arasında tanımlanmış ilişki, klasik kümelerin kartezyen çapımında olduğu gibi gerçekleştirilir. $X_1, X_2, \dots, X_n$ bulanık kümeleri tanımlansın. Bu n kümenin birbiri arasındaki ilişkiden elde edilecek

kümeleri tanımlansın. Bu n kümenin birbiri arasındaki ilişkiden elde edilecek kümenin elemanları $(X_1, X_2, \dots, X_n)$ elemanlarından oluşmuş olup üyelik dereceleri de klasik kümeye göre bir değişime uğramıştır. Değişime uğrayan üyelik dereceleri $[0-1]$ kapalı aralığında herhangi bir değere eşittir. Bu değerler kümeler arasındaki etkileşimin derecesini gösterir.

Örnek 2.5:

X ve Y bulanık kümeleri aşağıdaki gibi verilmiştir.

$$X = \{ \text{Newyork city, Paris} \}$$

$$Y = \{ \text{Beizing, Newyork city, London} \}$$

Bu iki küme arasındaki bulanık ilişki aşağıdaki gibi tanımlanmıştır.

$$R(x,y) = \{ 1/(Ncy, Be.) + 0.6/(Ncy, Lon.) + 0.9/(Pa., Be.) + 0.7/(Ncy, Pa.) + 0.3/(Lon., Pa.) \}$$

Bu ilişki, iki boyutlu bir matris olarak aşağıdaki gibi gösterilmiştir.

	<i>Ncy</i>	<i>Paris</i>
<i>Beizing</i>	1	0.9
<i>Ncy</i>	0	0.7
<i>London</i>	0.6	0.3

Klasik ilişki kapsamında düzenlenen kompozisyonlar, bulanık ilişkilerde de geçerlidir. Burada iki temel metot üzerinde durulacaktır.

2.4.1. Max - Min Kompozisyonu

$P(x, y)$, $Q(y, z)$, $R(x, z)$ üç ayrı bulanık kümeyi gösterebilir. Bu kümelerin elemanları aşağıdaki gibi tanımlanmıştır.

$$X = \{X_i / i \in I\}$$

$$Y = \{Y_j / j \in J\}$$

$$Z = \{Z_k / k \in K\}$$

P , Q ve R 'nin üyelik matrisleri şu şekilde gösterilmiştir.

$$P = [P_{i,j}], Q = [q_{j,k}] \text{ ve } R = [r_{i,k}]$$

P_{ij} , q_{jk} , r_{ik} üyelik dereceleri cinsinden aşağıdaki gibi gösterilmiştir.

$P_{i,j} = \mu_P(X_i, Y_j)$, $Q_{j,k} = \mu_Q(Y_j, Z_k)$, $R_{i,k} = \mu_R(X_i, Z_k)$ matrisleri içerisindeki her bir elemanın değeri $[0-1]$ kapalı aralığında değişecektir. Yukarıda bahsedilen P ve Q iki ayrık bulanık kümeyi, R ise bu iki bulanık kümenin ilişkisini göstermektedir. O halde R bulanık kümesi $P \circ Q = R$ ifadesiyle gösterilebilir. P ve Q arasındaki bu ilişki, max-min kompozisyonuna göre şu şekilde ifade edilmektedir.

$$R_{i,k} = \max_j \min [P_{i,j}, Q_{j,k}] \quad (2.31)$$

Örnek 2.6:

P ve Q matrisleri aşağıdaki gibi tanımlanmıştır.

$$P = \begin{bmatrix} 0.3 & 0.5 & 0.8 \\ 0 & 0.7 & 1 \\ 0.4 & 0.6 & 0.5 \end{bmatrix} \quad Q = \begin{bmatrix} 0.9 & 0.5 & 0.7 & 0.7 \\ 0.3 & 0.2 & 0 & 0.9 \\ 1 & 0 & 0.5 & 0.5 \end{bmatrix}$$

Bu iki küme arasındaki bulanık ilişki, max - min kompozisyonu ile aşağıdaki gibi yapılmıştır.

$$R [r_{i,k}] = \begin{matrix} & \begin{matrix} x & y & z \end{matrix} \\ \begin{matrix} x \\ y \\ z \end{matrix} & \begin{bmatrix} 0.3 & 0.5 & 0.8 \\ 0 & 0.7 & 1 \\ 0.4 & 0.6 & 0.5 \end{bmatrix} \end{matrix} \quad \text{or} \quad \begin{matrix} & \begin{matrix} x & y & z & w \end{matrix} \\ \begin{matrix} x \\ y \\ z \end{matrix} & \begin{bmatrix} 0.9 & 0.5 & 0.7 & 0.7 \\ 0.3 & 0.2 & 0 & 0.9 \\ 1 & 0 & 0.5 & 0.5 \end{bmatrix} \end{matrix}$$

Buradan elde edilecek $R [r_{i,k}]$ matrisi, 3x4 boyutlu bir matris olacaktır.

$$R [r_{i,k}] = \begin{bmatrix} r_{11} & r_{12} & r_{13} & r_{14} \\ r_{21} & r_{22} & r_{23} & r_{24} \\ r_{31} & r_{32} & r_{33} & r_{34} \end{bmatrix}$$

$$r_{11} = \max [\min (0.3, 0.9), \min (0.5, 0.3), \min (0.8, 1)]$$

$$r_{11} = \max [0.3, 0.3, 0.8] \Rightarrow r_{11} = 0.8$$

$$r_{12} = \max [\min (0.3, 0.5), \min (0.5, 0.2), \min (0.8, 0)]$$

$$r_{12} = \max [0.3, 0.2, 0] \Rightarrow r_{12} = 0.3$$

$$r_{13} = \max [\min (0.3, 0.7), \min (0.5, 0), \min (0.8, 0.5)]$$

$$r_{13} = \max [0.3, 0, 0.5] \Rightarrow r_{13} = 0.5$$

$$r_{14} = \max [\min (0.3, 0.7), \min (0.5, 0.9), \min (0.8, 0.5)]$$

$$r_{14} = \max [0.3, 0.5, 0.5] \Rightarrow r_{14} = 0.5$$

$$R [r_{i,k}] = \begin{bmatrix} 0.8 & 0.3 & 0.5 & 0.5 \\ 1 & 0.2 & 0.5 & 0.7 \\ 0.5 & 0.4 & 0.5 & 0.6 \end{bmatrix}$$

2.4.2. Max Çarpım Kompozisyonu

$P(x, y)$, $Q(y, z)$ ve $R(x, z)$ üç ayrı bulanık kümeyi gösterebilir. Bu kümelerin elemanları aşağıdaki gibi tanımlanmıştır.

$$X = \{ X_i / i \in I \}$$

$$Y = \{ Y_j / j \in J \}$$

$$Z = \{ Z_k / k \in K \}$$

P , Q ve R 'nin bir matris olarak ifadesi ve bu matris içerisindeki elemanların üyelik dereceleri aşağıda gösterilmiştir.

$$P = [p_{i,j}], Q = [q_{j,k}], R = [r_{i,k}]$$

$$P_{ij} = \mu_P(x_i, y_j), Q_{jk} = \mu_Q(y_j, z_k), R_{ik} = \mu_R(x_i, z_k)$$

Yukarıda bahsedilen P ve Q iki ayrı kümeyi, R ise bu iki bulanık kümenin ilişkisini gösterir. O halde bu bulanık ilişki $R = P \circ Q$ ifadesiyle gösterilebilir.

P ve Q arasındaki bu ilişki max çarpım kompozisyonu ile şu şekilde ifade edilebilmektedir. (Klir and Folger, 1988)

$$R_{i,k} = \max [P_{ij} \times Q_{j,k}] \quad (2.32)$$

Örnek 2.7:

P ve Q matrisleri aşağıdaki gibi tanımlanmıştır.

$$P = \begin{bmatrix} 0.5 & 0.6 \\ 0.8 & 0.7 \end{bmatrix} \quad Q = \begin{bmatrix} 0.6 & 0.5 \\ 0.2 & 0.8 \end{bmatrix}$$

Bu iki küme arasındaki bulanık ilişki max çarpım kompozisyonu ile aşağıdaki gibi yapılmıştır.

$$R[r_{i,k}] = \begin{bmatrix} 0.5 & 0.6 \\ 0.8 & 0.7 \end{bmatrix} \circ \begin{bmatrix} 0.6 & 0.5 \\ 0.2 & 0.8 \end{bmatrix}$$

Buradan elde edilecek $R[r_{i,k}]$ matrisi, 2x2 boyutlu bir matristir.

$$R[r_{i,k}] = \begin{bmatrix} r_{11} & r_{12} \\ r_{21} & r_{22} \end{bmatrix}$$

$$r_{11} = \max [(0.5 \times 0.6), (0.6 \times 0.2)]$$

$$r_{11} = \max [0.3, 0.12] \Rightarrow r_{11} = 0.3$$

$$r_{12} = \max [(0.5 \times 0.5), (0.6 \times 0.8)]$$

$$r_{12} = \max [0.25, 0.48] \Rightarrow r_{12} = 0.48$$

$$r_{21} = \max [(0.8 \times 0.6), (0.7 \times 0.2)]$$

$$r_{21} = \max [0.48, 0.14] \Rightarrow r_{21} = 0.48$$

$$r_{22} = \max [(0.8 \times 0.5), (0.7 \times 0.8)]$$

$$r_{22} = \max [0.4, 0.56] \Rightarrow r_{22} = 0.56$$

$$R[i,k] = \begin{bmatrix} 0.3 & 0.48 \\ 0.48 & 0.56 \end{bmatrix}$$

2.5. Bulanık Kümenin Türevleri

Bulanık kümelerin türevleri, bulanık bir kümeden klasik kümeler elde edileceği zaman kullanılmaktadır. Bir bulanık kümeye ait türevleri elde etmek için gerekli olan kavramları şu şekilde sıralanabilir.

2.5.1. α - Kesişim

Bir bulanık kümede üyelik fonksiyonunun α ' ya eşit yada α ' dan büyük değerlerine karşılık gelen elemanların oluşturduğu kümeye α - Kesim (α - Düzey) kümesine denir. Bu gibi bir kümenin Matematiksel ifadesi aşağıdaki gibi gösterilmiştir.

$$B_{\alpha} = \{x / \mu_B(x) \geq \alpha \} \quad x \in X \quad (2.33)$$

Örnek 2.8:

A bulanık kümesi ; $A = \{(X_1, 0.7), (X_2, 0.2), (X_3, 0.8), (X_4, 0.6), (X_5, 0.9)\}$ olarak veriliyor. A kümesinin α - kesim kümesinin $\alpha=0.2$ ve $\alpha=0.6$ için değeri aşağıda hesaplanmıştır.

$\alpha = 0.2$ iken ;

$\mu_A(x_1), \mu_A(x_2), \mu_A(x_3), \mu_A(x_4), \mu_A(x_5) \geq 0.2$ olduğundan $A_{0.2}$ kümesi

$$A_{0.2} = \{X_1, X_2, X_3, X_4, X_5\}$$

$\alpha = 0.6$ iken ;

$\mu_A(x_1), \mu_A(x_3), \mu_A(x_4), \mu_A(x_5) \geq 0.6$ olduğundan $A_{0.6}$ kümeside ;

$$A_{0.6} = \{X_1, X_3, X_4, X_5\} \text{ şeklinde elde edilir.}$$

2.5.2. Destek (Support)

Bir bulanık kümede, üyelik fonksiyonunun sıfırdan büyük değerine karşılık gelen elemanların oluşturduğu kümeye destek kümesi denir.

$$\text{desA} = \{ X / \mu_{A(x)} > 0 \} \quad x \in X \quad (2.34)$$

Örnek 2.9:

A bulanık kümesi , $A = \{ (X_1, 0.6), (X_2, 0), (X_3, 0.3), (X_4, 0.8), (X_5, 1) \}$ olarak veriliyor. A kümesinin destek kümesi $\mu_{A(X_1)}, \mu_{A(X_3)}, \mu_{A(X_4)}, \mu_{A(X_5)} > 0$ olduğundan ;

$$\text{desA} = \{X_1, X_3, X_4, X_5\}$$

şeklinde elde edilecektir. $\mu_{A(X_2)} < 0$ olduğundan desA kümesinin elemanı değildir.

2.5.3. Normallik

Verilen A bulanık kümesinin yüksekliği 1 ise A bulanık kümesi normaldir denir. $\text{Max} (\mu_{A(x)}) = 1$ ise A kümesi normaldir. Eğer A kümesi normal değilse tüm elemanlar yüksekliğe bölünerek normalize edilir.

2.5.4. Yükseklik

Bir A bulanık kümesinin yüksekliği, küme içerisindeki en büyük üyelik derecesine eşittir. Matematiksel olarak aşağıdaki gibi ifade edilmektedir.

$$\text{Yük}(A) = \max (\mu_{A(x)}) \quad x \in X \quad (2.35)$$

Örnek 2.10:

$X = \{a, b, c, d\}$ ve bu evrendeki A bulanık kümesi $A = \{ 0.6/a + 0.8/b + 0.9/c + 0.2/d \}$ olarak tanımlanıyor. A bulanık kümesinin yüksekliği ;

$$\text{Yük}(A) = \max (0.6, 0.8, 0.9, 0.2)$$

$\text{Yük}(A) = 0.9$ olarak elde edilir.

2.5.5. Genişletme (Extension)

Bir X evreninde, $\mu_A(X)$ üyelik işlevi ile tanımlanmış A bulanık önermesini ele alalım. Y evreni başka bir evren olmak üzere; $X \times Y$ evreninde $\mu_{\text{GEN}} \{A\} (x, y) = \mu_A(x)$ üyelik fonksiyonu ile tanımlanmıştır. İki evrenli bulanık önermeye, X evrenindeki A bulanık önermesinin $X \times Y$ evrenine genişletilmesi denir. Bu kısaca $\text{GEN}_{X \times Y} \{A\}_x$ evrenler biliniyorsa $\text{GEN} \{A\}$ olarak gösterilir. Görüldüğü gibi (x, y) öneri çiftinin üyelik derecesi, X evrenindeki x önerisinin üyelik derecesine bağlıdır.

Örnek 2.11:

X ve Y evrensel kümeleri aşağıdaki gibi tanımlanmış olsun.

X_1 önerilerinin konusu : Deney ortamının sıcaklığı

Y_1 önerilerinin konusu : Deney ortamının basıncı

$X = \{\text{Alçak}, \text{Orta}, \text{Yüksek}\} = \{X_1, X_2, X_3\}$

$Y = \{\text{Alçak}, \text{Normal altı}, \text{normal}, \text{normal üstü}, \text{yüksek}\} = \{Y_1, Y_2, Y_3, Y_4, Y_5\}$

X veya Y evrensel kümesindeki her öneriye bir üyelik derecesi vererek, bu evrensel kümelerde bulanık önerme her (x, y) öneri çiftine verilen üyelik derecesi ile iki evrenli bulanık önerme tanımlanmış olacaktır.

X evreninde : $A_1 = (0, 0.3, 1) \rightarrow$ (Anlamı : Sıcaklık yüksek)

$A_2 = (0.2, 1, 0.2) \rightarrow$ (Anlamı: Sıcaklık Orta seviyede)

Y evreninde : $B_1 = (0, 0.2, 1, 0.1, 0) \rightarrow$ (Anlamı: Basınç normal seviyede)

$B_2 = (1, 0.5, 0, 0, 0) \rightarrow$ (Anlamı : Basınç alçak seviyede)

$$R1 = \begin{bmatrix} 0.2 & 0.9 & 1 & 0.9 & 0.9 \\ 0 & 0.1 & 0.7 & 0.6 & 0.3 \\ 0 & 0.8 & 1 & 0.4 & 0.8 \end{bmatrix}$$

Bu önerme, $X \times Y$ evreninde tanımlanmış bir bulanık önermeyi ifade etmektedir.

$$R2 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0.3 & 0.3 & 0.3 & 0.3 & 0.3 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

X evrenindeki A_1 bulanık önermesinin, $X \times Y$ evrenine genişletilmiş bulanık önermesi (Anlamı : Sıcaklık Yüksek Basınç Önemsiz.)

$$R3 = \begin{bmatrix} 0 & 0.2 & 1 & 0.1 & 0 \\ 0 & 0.2 & 1 & 0.1 & 0 \\ 0 & 0.2 & 1 & 0.1 & 0 \end{bmatrix}$$

Y evrenindeki B_1 bulanık önermesinin $X \times Y$ evrenine genişletilmiş bulanık önermesi (Anlamı : Basınç normal, Sıcaklık önemsiz.)

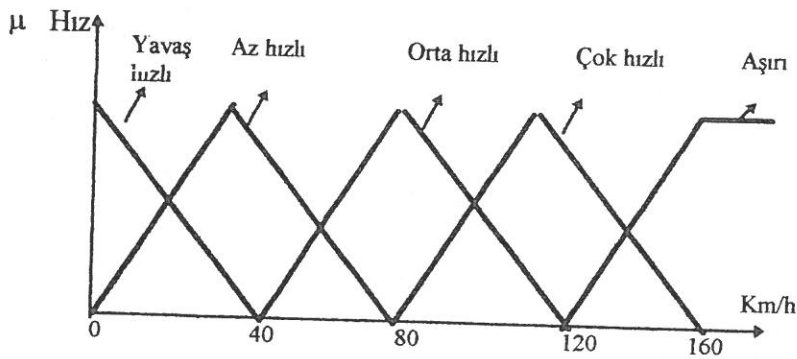
Yukarıdaki A_1 (sıcaklık yüksek) ve R_2 (sıcaklık yüksek, basınç önemsiz) önermeleri aynı fiziksel ölçümü ifade etmektedir. Dolayısıyla bir evrende ifade edilmiş fiziksel durumun bir üst evredeki ifade edilebilmektedir. Benzer şekilde R_1 (Basınç normal) ve R_3 (Basınç normal,sıcaklık önemsiz) önermeleride aynı fiziksel durumun bir üst evredeki ifadeleridir. (Çiftçi, 1996)

Bazı durumlarda bir bulanık kümenin üyelik fonksiyonlarının yerine, o kümenin elemanları gerekebilir. Yukarıda anlatılan kavramlar vasıtasıyla bulanık durumdaki bir kümenin klasik küme durumuna getirilmesi sağlanmaktadır.

3. SÖZEL DEĞİŞKENLER

3.1 Giriş

Bulanık kümelerde dilsel değişkenler birden fazla kelime yada kelime guruplarına sahiptir. Genel olarak bu değişkenler kelime yada kelime gurupları ile ifade edilmektedirler. Bu kelime veya kelime gurupları, sayılar kadar kesin bir anlam ifade etmediği için sınırları da bulanık olmaktadır. Günlük hayatımızda geniş bir uygulama alanına sahip olan bu kavramı bir örnekle açıklayalım. Yolda giden bir arabanın sürücüsünün önüne bir engel çıktığı zaman nasıl hareket eder? Burada girdilerimiz hız ve uzaklık çıktımız fren olmaktadır. Hızı 0-160 km/h arasında olan bir otomobili ele alalım. Hız değişkenleri sözel olarak konuşulduğu zaman yavaş, az hızlı, orta hızlı gibi değerler almaktadır. Bu değerler ise bahsedilen aralıktaki değerlere göre bir dereceye karşılık gelmektedir. Örneğimize göre hız için yavaş, az hızlı, orta hızlı, çok hızlı ve aşırı hızlı olmak üzere 5 ayrı bulanık küme seçilmiştir. Bunların her biri belli aralıkları derecelendiren sözel ifadelerdir. Hıza ait derecelendirme işlemi Şekil 3.1' de gösterilmektedir. (Zadeh, 1975; Günal, 1997)



Şekil 3.1 Hız değişkenine ait hız kümeleri

Görüldüğü gibi 0-160 km/h sınırları içerisinde olan hız değişkeni, belli aralıklar halinde derecelendirilerek sözel değişken durumuna dönüştürülmüştür. (0-40 yavaş, 0-80 az hızlı gibi)

3.2. Söz Dizimi (Syntax)

Bir sözel değişken X , $T(x)$, U , G , M şeklinde verilen beşli bir terimle ifade edilebilmektedir. Burada X değişkenin ismini, U değişkenin ait olduğu evreni, $T(x)$ sözel değerler kümesini, G söz dizimsel kuramı ve M her bir X ' in anlamı ile ilişkili G ile üretilen şematik kuralları göstermektedir. Fonksiyonu birim olan bir yada daha fazla kelimedenden oluşan terime “atomik terim” denir. Herhangi bir terim bir veya daha fazla terimi kapsıyor ise bu tür terimlere “bileşik terim” denir. Bileşik teriminin parçalarına da “alt terim” adı verilir.

Örnek 3.1:

Hız değişkenini ele alalım. $X = \text{Hız}$, $U = (0-160)$ hız sınırı temsil etmektedir. Hıza ait terim kümesi, $T(\text{hız}) = \{\text{yavaş, az hızlı, hızlı, çok hızlı, aşırı hızlı}\}$ şeklinde verilmiş olsun. Veriler içerisinde, hızın sözel değeri “hızlı” atomik bir terimdir. Bir başka terim “çok hızlı” bileşik terimdir. Bileşik terim içerisindeki “çok” ve “hızlı” terimleri, bileşik terimin birer alt terimleridir.

3.2.1 Yoğunluk

A bulanık kümesinin yoğunluğu, $YOG(A)$ ile gösterilir ve Üyelik fonksiyonunun karesi alınarak bulunur.

$$\mu_{YOG(A)}(x) = (\mu_A(x))^2 \quad x \in X \quad (3.1)$$

3.2.2 Genişlik

A bulanık kümesinin genişliği, $GEN(A)$ ile gösterilip yoğunluğunun tersi alınarak elde edilir.

$$\mu_{GEN(A)}(x) = 2\mu_A(x) - (\mu_A(x))^2 \quad (3.2)$$

3.2.3 Zıt Yoğunluk:

A bulanık kümesinin zıt yoğunluğu, $Z_{YOG(A)}(x)$ ile gösterilir ve aşağıdaki gibi hesaplanır.

$$\begin{aligned} \mu_{Z_{YOG(A)}}(x) &= 2[\mu_A(x)]^2 & 0 \leq \mu_A(x) < 0.5 \\ &= 1 - 2[1 - \mu_A(x)]^2 & 0.5 \leq \mu_A(x) < 1 \end{aligned} \quad (3.3)$$

ifadesi ile elde edilir.

3.3 Sözel Niteleyiciler

Doğal dilde kullanılan niteleyicileri kullanarak, sözel değişkenlerin değer aralığını genişletebiliriz. Çok , daha çok, az , daha az, yaklaşık olarak, zayıf ince gibi zarflara sözel niteleyiciler adını veririz. Bu niteleyiciler, daha çok kapalı anlamların anlaşılması için kullanılmaktadır. Örnek olarak “genç adam” önermesini ele alalım. Bu önermede sınırlar belli değildir. Ne kadar genç gibi bir soru aklımıza gelmektedir. Oysa bu önerme “çok genç adam” şeklinde olmuş olsaydı böyle bir soru aklımıza gelmeyecekti. Nedeni ise bu önermede gençliğin detayının belirtilmiş olmasından dolayıdır. Bir sözel niteleyici, bulanık kümelenirinin içerisindeki üyelik fonksiyonlarında yapılan işlemlerde olduğu gibi modellenir.

Sözel değişkenlerin değeri, çoğu zaman atomik terimlerin bir kompozisyonundan oluşmaktadır. Temel olarak atomik terimler aşağıdaki gibi incelenebilir :

- 1) Ön terimler: Evrensel kümenin bulanık alt kümelerine verilen addır. Buma örnek olarak genç, yaşlı, uzun, kısa gibi kavramları verebiliriz.
- 2) Birleştiriciler: Bir önermenin bağlaç görevi yapan terimleridir. Örnek olarak ve, veya, değil gibi bağlaçları verebiliriz.
- 3) Niteleyiciler: Bir önermede ön terimleri niteleyen terimlerdir. Örnek olarak çok biraz, pek gibi zarfları verebiliriz.
- 4) İşaretler: Parantez, artı, eksi gibi işaretleri içine alır. Örnek olarak terim kümesi $T(x)$ olan yaş değişkenini ele alalım.

$T(x) = \text{genç} + \text{yaşlı} + \text{çok genç} + \text{genç değil} + \text{pek yaşlı} + \text{çok çok yaşlı} + \text{daha genç} + \text{az çok genç}$

Burada belirtilen her bir terim, $U[0-150]$ aralığında verilen evrensel kümenin bir alt kümesidir. x , $T(x)$ terim kümesinin bir elemanı ise, bunun anlamı x , U evrensel kümesinin de bir alt kümesidir. Yukarıda verilen $T(x)$ terim kümesinin elemanlarından “genç” ve “Yaşlı” terimleri birer ön terimi teşkil ederken, diğer elemanlar atomik terimlerin birer kombinasyonudur.

Dilbilgisi kurallarına göre birbirine bağlanarak elde edilen kelime guru- buna söz dizimi adı verilir. Bunlar genellik isim-sıfat şeklinde temsil edilirler. İlk olarak bir T söz diziminin alt kümesi ile işe başlayacağız. Söz dizimi için tanımlanmış bir U evresinde, her zaman için bir sıfat gurubu vardır. Buna örnek olarak Genç (adam), süratli (araba), büyük (ev) gibi ifadeler verilebilir. T_q , U evreninde tanımlanmış ve T sözdiziminin alt kümesi olan bir

bulanık alt kümeyi, m ve v U evreninde belirlenmiş olduğumuz noktaları ($m \leq v$) göstermek üzere; $\text{des}(\mu(A)) \subseteq \langle m, v \rangle$ ifadesi elde edilir. Ayrıca söz dizimlerini düzenleyebilmek için, $m \leq s \leq v$ şartını sağlayan bir s noktası seçilir. s noktasına U' nun merkez noktası adı verilir. m, s, v noktaları dikkate alınarak söz dizimleri düzenlenirse;

- 1) $\text{Nor}(\mu(A^+)) \subseteq \langle s, v \rangle \Rightarrow A^+$
- 2) $\text{Nor}(\mu(A^-)) \subseteq \langle m, s \rangle \Rightarrow A^-$
- 3) $A \in T_q$ söz dizimi için; $\text{se Nor}(\mu(A)) \Rightarrow A^0$

sonuçları elde edilmektedir. Burada A^- negatif söz dizimini, A^+ pozitif söz dizimini, A^0 ise sıfır söz dizimini göstermektedir. Örnek olarak T söz diziminin elemanları büyük (ev), küçük (ev) ve orta (ev) şeklinde verilmiş olsun. Burada A^+ yı büyük (ev), A^- yi küçük (ev) ve A^0 ı orta (ev) olarak göstermek mümkündür. (Kuz'min, 1985)

Bir sözel niteleyicinin üyelik fonksiyonu, Atomik terimin üyelik fonksiyonundan faydalanarak hesaplanmaktadır. Bunlardan bazılarının hesaplanması aşağıda gösterilmiştir. (Zadeh, 1976)

a) Çok sözel niteleyicisi: Çok sık kullanılan bu niteleyicinin değeri aşağıda gösterilmiştir.

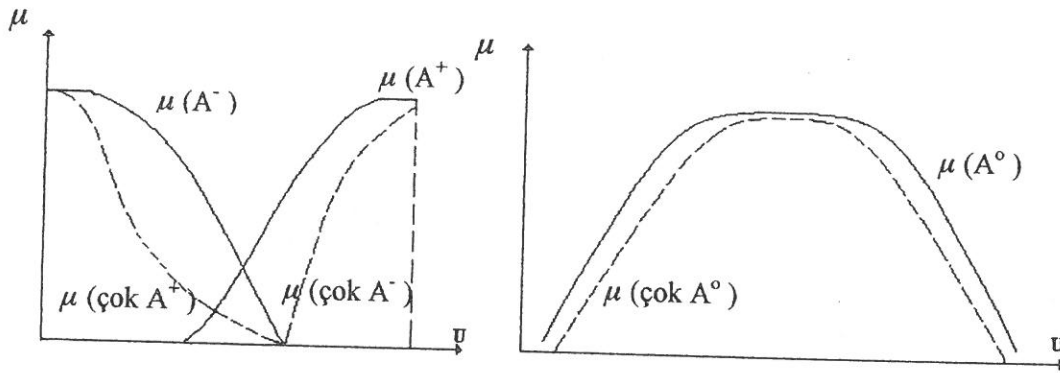
$$V_{\text{çok}}(\alpha) = \text{YOG}(\alpha) \quad \alpha \in L \quad (3.4)$$

$$\tau_{\text{çok}}(x) = \begin{cases} y, & y \leq x \text{ A pozitif veya sıfır ise} \\ y, & y \geq x \text{ A negatif veya sıfır ise} \end{cases}$$

y değeri, A' 'nın normaliyle tanımlanmış ise $\tau_{\text{çok}}(x)$ değerine eşit olur ve $\tau_{\text{çok}}(x)$ değeri aşağıdaki gibi ifade edilir.

$$\tau_{\text{çok}}(x) = x + (-1)^k d \cdot \text{Nor}(A) \quad (3.5)$$

Burada A , pozitif veya sıfır iken $k=1$ ve A negatif veya sıfır iken $k=2$ dir. d ise, $0 \leq d \leq 0.5$ aralığında tanımlanmış bir sabiti göstermektedir. Şekil 3.2' de çok niteleyicinin sonucunda elde edilen grafikler gösterilmiştir.



Şekil 3.2. Çok niteleyicisinin yorumlanması

Örnek 3.2:

U , A kümeleri aşağıdaki gibi tanımlanmıştır. (A , U evreninde tanımlanmış bir bulanık kümeyi göstermektedir.)

$A = \text{küçük}$,

$U = \{1, 2, \dots, 10\}$,

$M(\text{küçük}) = \{1/1, 1/2, 0.8/3, 0.6/4, 0.5/5, 0.4/6, 0.2/7\}$,

verilenlere göre $m = 1$, $v = 7$ ve semantik merkez $s = 5$ olarak yazılabilir.

Nor (küçük) = {1, 2} bu değerleri $<1, 5>$ aralığında kaldığı için bu söz dizimi negatiftir.

Buradan;

$$M (\text{çok küçük}) = \{1/1, 1/2, 0.64/3, 0.36/4, 0.25/5, 0.16/6, 0.04/7\}$$

b) Pek çok niteleyicisi: Bu niteleyicide çok niteleyicisine benzer bir şekilde elde edilir.

$$V_{\text{pek çok}}(\alpha) = \alpha.\alpha.\alpha \quad \alpha \in L \quad (3.6)$$

Yukarıdaki örnek için;

$$\mu (\text{pek çok küçük}) = \{1/1, 1/2, 0.512/3, 0.216/4, 0.125/5, 0.064/6, 0.008/7\}$$

olarak elde edilir.

c) Daha çok veya daha az niteleyiciler: Bu niteleyici çoğu zaman, çok niteleyicisinin tersi olarak bilinir. Değeri aşağıdaki gibi hesaplanmaktadır.

$$V_{\text{daha çok veya daha az}}(\alpha) = \text{GEN}(\alpha) \quad \alpha \in L \quad (3.7)$$

$$\tau_{\text{daha çok veya daha az}}(x) = \begin{cases} y \geq x & \text{Eğer A pozitif veya sıfırsa} \\ y \leq x & \text{Eğer A negatif veya sıfırsa} \end{cases}$$

$$\tau_{\text{daha çok veya daha az}}(x) = x + (-1)^k d \text{ Nor}(A) \quad (3.8)$$

Yukarıdaki eşitlikte Nor (A) için iki yaklaşım söz konusudur. İlkinde A sıfır veya negatif iken $k=1$ ' dir. İkincisinde A sıfır veya pozitif iken $k=2$ ' dir.

Örnek 3.3:

U ve A kümeleri örnek 1' deki gibi tanımlanmış olsun. Bu durumda

$$\mu_{\text{(daha çok veya daha az küçük)}} = (1/1, 1/2, 1/3, 0.96/4, 0.84/5, 0.75/6, 0.64/7, 0.36/8)$$

olarak elde edilir.

d) Kabaca niteleyicisi : Bu niteleyici aşağıdaki formül ile elde edilir.

$$V_{\text{kabaca}}(\alpha) = \text{GEN}(\text{Gen}(\alpha)), \quad \alpha \in L \quad (3.9)$$

$L = <0,1>$ aralığı için V_{kabaca} niteleyicisi aşağıdaki gibi elde edilir.

$$V_{\text{kabaca}}(\alpha) = -\alpha^4 + 4\alpha^3 - 6\alpha^2 + 4\alpha \quad (3.10)$$

τ_{kabaca} fonksiyonu, $\tau_{\text{daha çok}}$ 'taki forma benzer bir şekilde elde edilmiştir. Burda A pozitif için, $\tau_{\text{kabaca}} \geq \tau_{\text{daha çok}}$ veya daha az ve A negatif için $\tau_{\text{kabaca}} \leq \tau_{\text{daha çok}}$ veya daha az olarak kabul edilecektir. U ve A' nın değeri örnek 3.1 deki olduğu gibi düşünülürse;

$$\mu_{\text{(kaba küçük)}} = (1/1, 1/2, 1/3, 0.998/4, 0.974/5, 0.938/6, 0.87/7, 0.59/8)$$

olarak elde edilir.

e) Oldukça niteleyicisi: Bu niteleyici aşağıdaki gibi tanımlanmıştır.

$$V_{\text{oldukça}} = M_{\text{ZYOG}}(\text{YOG}(\alpha)), \quad \alpha \in L \quad (3.11)$$

"Eş. 3.10" $L = <0,1>$ aralığı için düzenlenirse;

$$V_{\text{oldukça}}(\alpha) = (2\alpha^4 \wedge \alpha^2) \cup (-2\alpha^4 + 4\alpha^2 - 1) \quad (3.12)$$

olarak elde edilir. U ve A örnek 3.1' deki gibi alınırsa ;

$$\mu_{\text{(oldukça küçük)}} = (1/1, 1/2, 0.74/3, 0.259/4, 0.125/5, 0.051/5, 0.003/7)$$

sonucu elde edilecektir.

4. ÜYELİK FONKSİYONLARI

4.1. Giriş

Bulanık kümelerde, üyelik fonksiyonları çok önemli bir yere sahiptir. Bilindiği üzere; bulanık küme teorisinin iki temel özelliği vardır. Bunlardan birincisi; üyelik fonksiyonlarının seçiminde operatörün en uygun olanını seçmesi, ikincisi ise bulanık küme teorisinin esnek bir yapıya sahip olmasıdır. Birincisi unsur dikkate alındığında; Bulanık küme teorisi gerçek bir hayat problemlerine uygun bir şekilde seçilmesi gerekecektir.

Genel olarak bir üyelik fonksiyonu iki şekilde seçilebilir. Birincisi tercih tabanlı olarak yapılan seçim, ikincisi olabilirlik dağılımlı yapılan seçimdir. Bulanık kümede insani faktörler dikkate alındığından, üyelik fonksiyonlarını seçerken sübjektif bir yapıda olmasına dikkat etmek gerekir.

Üyelik fonksiyonlarını belirlerken dikkat edilmesi gereken diğer bir hususta yapılan ölçüm düzeyidir. Bazı bilim adamları insani faktörlerin ve sözel değişkenlerin, problemin çözümüne daha fazla katılabilmesi için ölçme düzeyinin düşük olmasını isterken, diğer taraftan matematikçiler işlemlerin anlamlı olabilmesi için ölçme düzeyinin yüksek olması istemektedir.

4.2. Üyelik Fonksiyonlarının Seçiminde Kullanılan Metotlar

4.2.1. Frekans yaklaşımı

Üyelik fonksiyonları değerlerinin açık bir şekilde hesaplanması için kullanılmıştır. Bu yaklaşımda değerler, master haline getirilerek

çözölebilmektedir. Bu konuda bir gurup arařtırımacı, kendilerine ařağıdaki gibi bir soru yönelıtmıř ve cevabını vermiřlerdir.

X_0 ile temsil eden genel kavramlar, A gibi bir bulanık kümeyle gösterilebilir mi? Burada X_0 , sonlu bir evrensel kümenin elemanlarıdır. Bu soruya “evet” yada “hayır” řeklinde olacaktır. Verilen cevap olumlu ise “yes” üyelik fonksiyonu ařağıdaki gibi hesaplanır.

$$\mu_A(x_0) = \frac{n(X_0)}{N} \quad (4.1)$$

Burada $n(X_0)$ evrendeki “evet” cevaplarının sayısını, N ise toplam cevap sayısını (“evet” ve “hayır” ların toplamı) gösterir. Bu istatistiki yaklařım ile olabilirlik seviyesinde hesaplanabilir. Alt sınır deęeri $\mu^-_A(x_0)$ ve üst sınır deęeri $\mu^+_A(x_0)$ bu yaklařımla hesaplanabilir ve böylece hesaplanan üyeliklerin deęeri, olması gereken deęere yakın bir řekilde elde edilmiř olur.

4.2.2. Uzaklık yaklařımı

Bu yaklařım, bir referans noktasından $d(x)$ kadarlık bir uzaklıęı hedef seęer ve çoęu alanda kullanılır. Örneęin çok terimli karar verme, çok amaçlı karar verme programlarında olduęu gibi Zimman ve Zysno bu kavramı üyelik fonksiyonu hesaplanmasında kullandılar. Bu kavramın esası řöyledir. Nesne, refrans olarak seęilen nokta ile aynı özelliklere sahipse aralarındaki uzaklık $d(x)$ sıfır, nesne ile referans noktası arasında hię bir benzerlik yoksa aralarındaki uzaklık $d(x)$ sonsuz olacaktır. Bu düşünce evrende tanımlanmıř X bulanık kümesi için tanımlanırsa, $x \in X$ kümesinde seęilmiř herbir x elemanı için, bir üyelik derecesi tanımlamak mümkündür. Zimman ve Zysno μ ve $d(x)$ arasında iliřkiyi ařağıdaki gibi önermektedir.

$$\mu = \frac{1}{(1 + d(x))} \quad (4.2)$$

$d(x)=0$ iken $\mu=1$; $d(x)=\infty$ iken $\mu=0$ olmaktadır. Bu dönüşüm, R gerçekte sayısını $[0 - 1]$ aralığında taşır. yapılan deneysel çalışmalar, konuya ait kavramlar ve uzaklık fonksiyonunun konuya bağımlı olduğunu göstermektedir. Konuya bağımlı parametreler a ve b dir. a ; metre, yada , ayak gibi uzunluk birimidir, b ise daha pozitif daha negatif gibi değerler alabilir. Zimman ve Zysno uzaklık fonksiyonu ve konuya bağımlı parametreler arasındaki bağıntıyı aşağıdaki gibi tanımladılar.

$$\begin{aligned} d(x) &= \frac{1}{e^{-a(x-b)}} \\ \mu &= \frac{1}{[1 + e^{-a(x-b)}]} \end{aligned} \quad (4.3)$$

a parametresi üyelik fonksiyonunda eğilimi uygun sınırlar içinde modelleme b parametresi ise nesne davranışını pozitiften negatife değiştirmekte kullanılır. Yukarıdaki son ifade, değişik sistemlerin üyeliklerinin modellenmesindeki genel bir eşitliktir. Fonksiyonunun alt ve üst sınırlar için seçilen değerleri sırasıyla 0 ve 1 dir. Çoğu zaman anlaşılan durumu temsili için lojistik fonksiyonun bir parçası gereklidir. Bu iyi bir kalibrasyonla gerçekleşir. Bu durumda aralık ölçüm düzeyinde iki uzaklığın oranı değişmez. μ^0 , μ_L ve μ_1 sırasıyla normalleştirilmiş üyelik ölçeğindeki üst sınır, alt sınır ve üyelik derecesini temsil eden değerler olsun. O zaman;

$$\frac{\mu_1 - \mu^L}{\mu^u - \mu^L} = \frac{\mu'_1 - \mu^L}{\mu^u - \mu^L}$$

Normalize edilmiş bir üyelik fonksiyonunda alt ve üst sınır değerleri;

$$\mu^L = 0, \quad \mu^u = 1 \quad \text{olduğundan};$$

$$\mu'_1 = \mu_1(\mu^u - \mu^L) + \mu^L \quad (4.4)$$

sonucu elde edilir. Genellikle, dar aralığı ve c merkezi belirlemede tercih edilen parametrelerdir.

$$d = \mu^u - \mu^L \quad \text{ve} \quad c = (\mu^u + \mu^L)/2$$

bu taktirde,

$$\begin{aligned} \mu'_1 &= \mu_1(d) + \mu^L \\ &= d\mu_1 - d/2 + d/2 + \mu^L \\ &= d\mu_1 - d/2 + [(\mu^u - \mu^L)]/2 + \mu^L \\ &= d(\mu_1 - 1/2) + c \end{aligned}$$

μ_1 için eşitlik yeniden düzenlenirse tam üyelik fonksiyonu aşağıdaki gibi elde edilir.

$$\mu_1 = \left[(1/d)(1/(1 + e^{-a(x-b)}) - c) + 1/2 \right] \quad (4.5)$$

4.2.3. Saaty' nin ikili karşılaştırmalar metodu ile üyelik fonksiyonlarının hesaplanması

Üyelik fonksiyonu oransal ölçekte Saaty' nin önerdiği gibi ikili karşılaştırmalar metodu ile kolaylıkla hesaplanabilir. Her zamanki gibi μ_i , kümenin i 'ninci elemanının üyelik derecesini gösterebilir. μ_i / μ_j oranıyla karesel bir matris oluşturur. Bu matris $\mu = [\mu_1 \mu_2 \mu_3 \dots \mu_n]$ vektörleri ile çarpılarak

$$[\mu_i / \mu_j] \mu = n \mu \quad (4.6)$$

eşitliği elde edilir.

$$([\mu_i / \mu_j] - n) \mu = 0 \quad (4.7)$$

Böylece en büyük özdeğer “n” ve buna karşılık gelen “μ” özvektörü elde edilmiş olmaktadır.

4.2.4. Olasılık karakteristikleri ile üyelik fonksiyonlarının hesaplanması

Daha önceki yöntemlerde bir olasılık fonksiyonu hesaplandı. Parçalı olasılık $P_1, P_2, P_3 \dots P_n$ $\sum p_i = 1$ için azalan sırada düzenlenirse ($P_1 \geq P_2 \geq P_3 \geq \dots \geq P_n$), üyelik fonksiyonunun $\mu_1, \mu_2, \dots, \mu_n$ değer aşağıdaki gibi formülize edilir.

$$\mu_i = 1 - \sum_{j=1}^{i-1} (P_j - P_i), i > 1 \quad (4.8)$$

“Eş. 4.8” normalleştirme dikkate alınarak ($\mu_i = 1$) düzenlenirse;

$$\mu_i = 1 - \sum_{j=i+1}^n P_j \quad (\text{kolaylaştırmak için } \sum_{j=n+1}^n = 0)$$

P_i değerlerinin hiç bir düzenleme yapılmadığı varsayarak;

$$\mu_i = \sum_{j=1}^n \text{Min} (P_i, P_j) \quad (4.9)$$

Kontrolde üyelik fonksiyonları ve buna karşılık gelen olasılık fonksiyonu aynı şekildedir. Bu yüzden,

$\mu = \mu_j$ iken $P_i = P_j$ ve $\mu_i \geq \mu_j$ iken $P_i \geq P_j$ olacaktır ve aşağıdaki gibi yazılacaktır.

$$P_i = \sum_{j=1}^n \frac{1}{j} (\mu_j, \mu_{j+1}, i=1,2,\dots,n \text{ ve } \mu_{n+1}=0) \quad (4.10)$$

R' de bir olasılık yoğunluk fonksiyonu (o.y.f) tanımlandığında, buna karşılık gelen üyelik fonksiyonu performans index' in minimumu alınarak hesaplanmaktadır.

$$\min_A \int A^2(x) dx$$

Nesneye ait kıstaslar ise;

$$1) E(A(x)) \mid x \text{ (o.y.f' ye göre ayarlanmış)}$$

Burada $E(.)$ A' nin üyelik fonksiyonunun beklenen değeri olup 1' ye yakın bir değer almaktadır.

$$2) 0 \leq A(x) \leq 1$$

Yukarıdaki formülün integrali alındığında A kümesinin gerçek değeri elde edilir. Optimizasyon tekniklerinin kullanılmasıyla sonsuz boyutlu bir uzay için üyelik fonksiyonları aşağıdaki gibi hesaplanmaktadır. (Petryez, 1989)

$$A_{(x)} = \begin{cases} \lambda_p(x), & \text{Eğer } \lambda_p(x) < 1 \\ 1, & \text{değilse} \end{cases} \quad (4.11)$$

Burada λ aşağıdaki eşitlik kullanılarak elde edilir.

$$\lambda = \int_{\lambda_P(x)} P^2(x) dx + \int_{\lambda_P(x) \geq 1} P(x) dx - c = 0 \quad (4.12)$$

5. BULANIK MANTIK DENETLEYİCİLERİ

5.1 Giriş

Denetleyici sistemlerin genelinde sistem faaliyetlerini temsil eden mantık, kontrol sistemine uygun bir şekilde olması gerekmektedir. Klasik kontrol teorisinde sistemin yapısını açıklayan bilgiler, kesin değerler halinde verilir. Kontrol stratejisinin temelini; sisteme ait bilgilerle sistem değişkenleri arasında ilişkiler oluşturur. Bununla birlikte her sistemin modelini tanımlamak her zaman kolay olmayacaktır. Böyle sistemlere klasik kontrol uygulanırsa genellikle düşük ve basit kapasiteli kontrol algoritmaları kullanılacaktır. Bu yüzden sistemin davranışı istenilen durum ve ayar noktasından sapar. Kontrol sisteminde meydana gelen problemlerin sebebi, teori ve uygulama arasındaki uyumsuzluklardır. Bunun çözümü ise, kontrol edilecek sistemde bulunan belirsizliğin uygun bir şekilde yorumlanmasında yatar. Bulanık mantık denetleyicilerin ortaya çıkması ile bu problemlerin bazılarına, özellikle belirsizlik problemlerine yeni çözümler bulunmuştur. Bulanık mantık sezgi ve deneyime dayalı bir kontrol algoritmasıdır.

Bu bölümde, bulanık mantık denetleyicileri hakkında bilgi verilecektir. (Baba, 1995)

5.2 Bulanık Mantık Denetleyicilerin Genel yapısı

Bulanık mantık, endüstriyel süreçleri denetlerken kesin ve tam sayısal kurallardan ziyade dilsel kuralları içine alır. Bulanık mantık denetleyicileri, klasik ve modern kontrol teorisinde olduğu gibi kesin ve tam matematik modellere ihtiyaç duymaz. Çoğu sistemde benzer model ölçümleri belirtmek oldukça zordur. Denetlemesi zor olan karmaşık süreçlerde (çimento ocakları,

çelik fırınları çöp işleme fabrikaları gibi), bulanık mantık denetimini kullanmak zorunlu hale gelmektedir. (Taşkın ve Gümüştas, 1996)

Bulanık mantık denetleyicilerle daha az bir çabayla daha fazla iş yapılabilir. Deneyimler etkin bir şekilde kullanılarak fiziksel bir sistemin kontrolü, aşağıdaki dört unsur dikkate alınarak yapılmaktadır.

1. Mikrodenetleyicilerle çıkarım işlemcisi kaskat bağlayıp beraber çalıştırmak,
2. Yazılım denetleyicisi kullanmak ,
3. Bilgisayar tabanlı uygulamalarda ise; kural tabanı, veri tabanı, bulandırıcı, çıkarım motoru ve berraklaştırıcı olarak yazılım kullanmak ve paralel iletişimle kontrol sistemini tasarlamak,
4. İçinde RAM, EPROM, I/O birimlerinin yanı sıra bulandırıcı, çıkarım motoru ve durulatici bölümlerinde bulunduğu tümdevre şeklinde bulanık işlemciler kullanarak fiziksel sistemlerin kontrolünü sağlamak mümkün olacaktır. (Doğueri, 1996)

Bulanık mantık denetleyicilerin dayandığı temel nokta; uzman bir sistem operatörünün bilgi deneyim sezgi ve kontrol stratejisini, denetleyici tasarımında bilgi tabanı olarak oluşturmaktadır. Kontrol işlemleri bilgi ve deneyime dayanan sözel kurallarla gerçekleştirilir. Örneğin bir uzman, sistem için gerekli olan kontrol davranışlarını “küçük”, “hızlı”, “yavaş” gibi sözel terimlerle tanımlarsa, “Eğer-ise” (If-Then) komutlarıyla oluşturulacak kurallarda sözel terimler kullanılarak elde edilecektir.

Şekil 5.1 de bir bulanık mantık denetleyicinin iç yapısı görülmektedir. Bu denetleyici , genel olarak dört ana kısımdan oluşur.

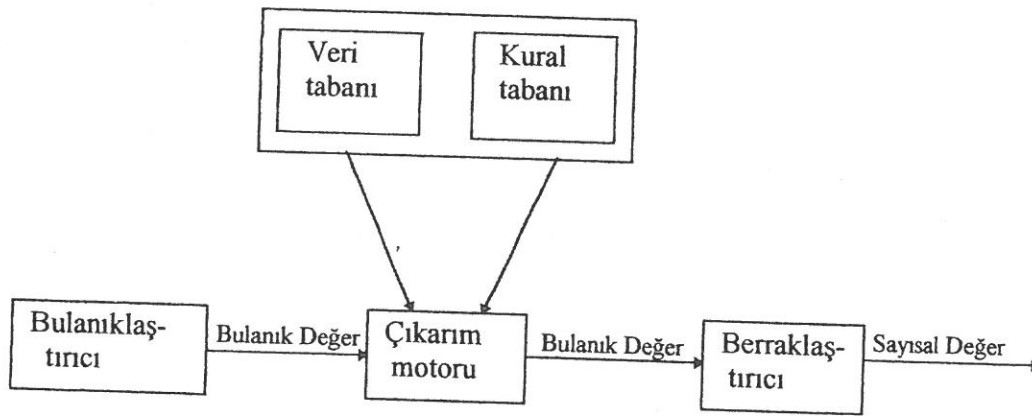
1) Bulanıklaştırma ünitesi (Fuzzifier): Bu bölüm giriş değişkenlerini ölçer, onlar üzerinde bir ölçek değişikliği yaparak bulanık kümelerle dönüştürür. Yani onlara bir etiket vererek, dilsel bir ölçek değişikliği yaparak bulanık mantık kümeleri haline dönüştürür.

2) Çıkarım motoru (Inference Engine): Bu ünite, kurallar üzerinde bulanık mantık kurallarını uygulayarak bulanık çıkışlar verir. Burada insanın düşünüş şeklinin benzetimini yapılmaya çalışılmıştır.

3) Veri tabanı (Data Base): Çıkarım motoru, kural tabanında kullanılan bulanık kümeleri bu bölümden alır.

4) Kural tabanı (Rule Base): Kontrol amaçlarına uygun dilsel denetim kuralları buradan bulunur ve çıkarım motoruna verilir.

5. Berraklaştırma (Defuzzifier): Çıkarım motorunun bulanık küme üzerinde yapmış olduğu ölçek değişikliklerini, sayısal değerlere dönüştürür.

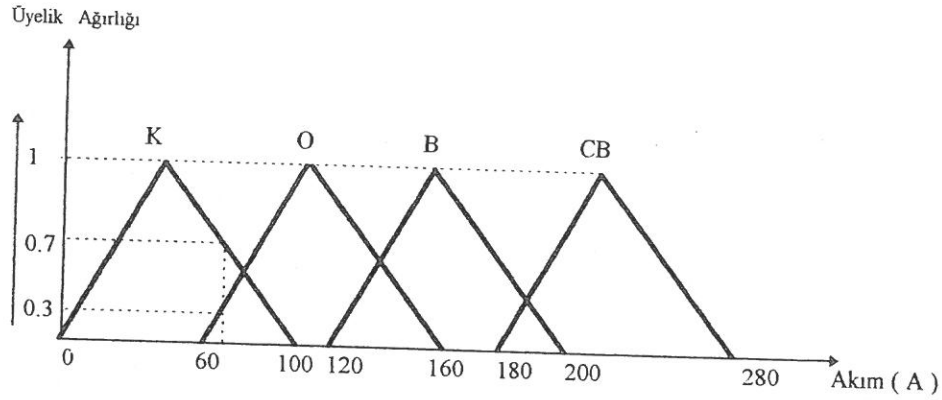


Şekil 5.1 Bulanık mantık denetleyicisinin blok diyagramı

5.2.1 Bulanıklaştırma Ünitesi

Bu ünite, sensörden alınan sayısal bilgileri bulanık formatta temsil etmektedir. Bilgilerin bulanık forma dönüştürülmesi, o büyüklüğün sıfır ile bir

arasındaki değerlerle ifade edilmesi demektir. Bu şekilde bir sayısal büyüklük sözel olarak ifade edilmiş olur. Sayısal büyüklükleri bulanık forma dönüştürebilmek için ilk olarak sabit mantık kümelerinin (Üniverse of Discourse) tanımlanması gerekir. Daha sonra sabit mantık kümeleri içerisindeki her bir değer için üyelik fonksiyonları ile tanımlanması gerekir. Üyelik fonksiyonları genellikle “sıfır (SF)”, “küçük pozitif (KP)”, “küçük negatif (KN)” gibi gruplarla temsil edilirler.



Şekil 5.2. Giriş değerlerinin bulanık forma dönüştürülmesi

Şekil 5.2’ de akım olarak verilen bir girişin bulanık forma dönüştürülmesi gösterilmiştir. Burada “K” küçük bulanık kümesini, “O” orta bulanık kümesini, “B” büyük bulanık kümesini ve “CB” çok büyük bulanık kümesini temsil etmektedir. Girişten 70 mA’lık bir değer geldiğinde, bu değer için bulanık formdaki yeni ismi “küçük bölgesinde üyelik ağırlığı 0.7” ve “orta bölgesinde üyelik ağırlığı 0.3” şeklinde olacaktır. (Kıray, 1997)

5.2.1.1 Üyelik Fonksiyonlarının Oluşturulması

Bulanık mantıkta, verilen bir sayısal ifadenin üyelik ağırlığı üyelik fonksiyonları vasıtası ile gerçekleştirilir. Üyelik fonksiyonunun tespiti oldukça

önemli bir vasıta olup, sistemin hassasiyetini belirtir. Üyelik fonksiyonları tespit edilirken özel bir yöntem kullanılmaz. İlk önce dilsel olarak ifade edilecek bölgenin sınırları tespit edilmesi gerekir. Bu sistemin en kaba hassasiyetini belirtir. Bu sistemin en kaba hassasiyetini belirtir. Daha sonraki hassasiyet ise, üyelik fonksiyonlarının şekilleriyle ayarlanır. En kullanışlı üyelik fonksiyonu elde edinceye kadar bir çok deneme yapılır. Üyelik fonksiyonlarının tespiti, uzman kişinin deney tecrübeleri sonucunda elde edilir.

5.2.2 Bilgi Tabanı

Bulanık mantık denetleyicilerinde bilgi tabanı kural tabanı ve veri tabanından oluşmaktadır. Kural tabanı; uzmanlar tarafından belirlenmiş, bulanık mantık denetleyicisinin davranışlarını tespit eden kontrol kurallarını içermektedir. Veri tabanı ise bulanık mantık denetleyicisinin kurallarını ve üyelik fonksiyonlarını tanımlamaktadır.

Kural tabanı, karar verme işleminde kullanılan birçok paralel kuraldan ve kontrol sistemi değişkenlerinden oluşmuştur. Bu kurallar, sistemin giriş-çıkışı arasındaki mantıksal ilişkileri belirlemektedir. Bulanık mantık denetleyicisinin çıkışı, bu kuralların değerlendirilmesi sonucunda elde edilmektedir. Kurallar sistem değişkenlerinin tanımladığı “Eğer” ve kontrol değişkenlerinin tanımladığı “ise” komutlarıyla oluşturulmaktadır.

Kural davranışı, bulanık kümelerle tanımlanmış kurallar aşağıdaki gibi ifade edilir:

$$\text{Eğer } X_1 ; A_1 \text{ ise Ve (Veya) } X_2 ; A_2 \text{ ise } U ; B_1 \quad (5.1)$$

Burada X_1 ve X_2 kontrol edilerek sistemin değişkenlerini oluşturur. U kurala ait kontrol çıkışını gösterir. X_1 , X_2 ve U değişkenleri sırasıyla A_1 , A_2 ve B_1 bulanık mantık elemanlarını göstermektedir.

“Eş. 5.1” kuralının davranışı, B_1 üyelik fonksiyonları yerine sayısal bir değerde yazılabilir.

$$\text{Eğer } X_1 \text{ ve } A_1 \text{ ise ve (veya) } X_2; A_2 \text{ ise } U = -0.50 \quad (5.2)$$

“Eş. 5.2” sözel bir örnekle açılanacak olursa; “eğer hız yüksek ve (veya) mesafe az ise kontrol davranışı -0.50 yap”, şeklinde ifade edilebilir.

“Eş. 5.1” deki kuralda B_1 üyelik fonksiyonu yerine $u = f(x_1, x_2)$ şeklinde sürekli bir fonksiyonda yazılabilir. Burada x_1 ve x_2 sistem değişkenlerini göstermektedir. Kuralı temsil eden kontrol davranışı u , x_1 ve x_2 değişkenlerinin bir fonksiyonu olarak gösterilmiştir. u kural davranışı, sistem içerisinde ayarlanabilen parametreler dikkate alındığında aşağıdaki gibi bulunacaktır.

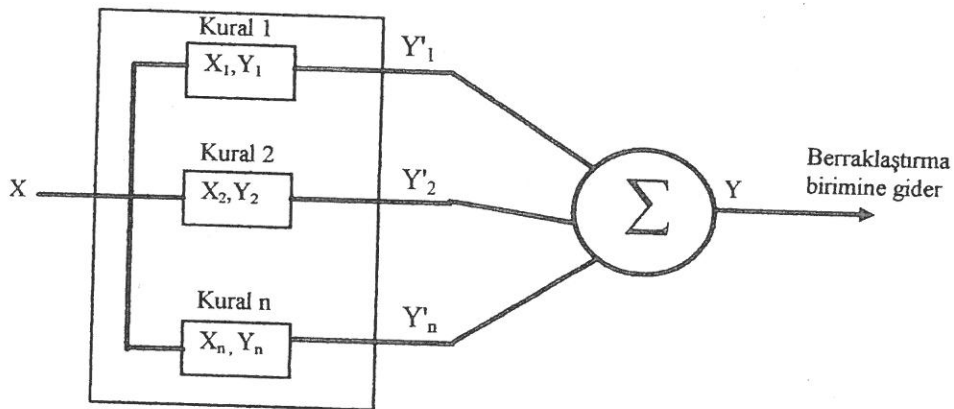
$$u = d_0 + d_1x_1 + \dots + d_nx_n$$

Kontrol alanında yapılan uygulamalarda, kontrol çıkışı genelde singleton bir fonksiyon olarak seçilir. Bir bulanık mantık denetleyicisinde, kontrol kuralları 4 farklı metot kullanılarak elde edilebilir.

- Kontrol kuralları doğrudan doğruya uzman kişinin bilgi ve deneyiminden yararlanılarak elde edilir.
- Kontrol kuralları sistemin karakteristiklerinin bulanık şekilde ifade edilmiş modellerden elde edilebilir.
- Operatörünün kontrol davranışları gözlenir ve denetleyici bu davranışlardan bulanık model oluşturur.
- Kurallar denetleyici tarafından gözetlenebilir.

Tek girişli bir sistem için n adet kuraldan oluşmuş kural tabanının yapısı Şekil 5.3' te gösterilmiştir. Burada X bulanık mantık denetleyicisinin girişini, Y bulanık denetleyicinin çıkışı, X_n, Y_n sırasıyla n . kurala ait giriş ve çıkış kümelerini, Y'_n n . kurala ait bulanık denetleyici davranışını ve W_n kurallarının skaler ağırlıklarını göstermektedir. Kuralların skaler ağırlıkları dikkate alındığında Y bulanık çıkışı aşağıdaki gibi elde edilecektir. (Baba, 1995)

$$Y = W_1 \cdot Y'_1 + \dots + W_n \cdot Y'_n \quad (5.3)$$



Şekil 5.3. Kural tabanı

5.2.3. Çıkarım Ünitesi

Bu ünite, Bulanıklaştırma biriminden gelen bulanık değerleri, kural tabanındaki kurallar üzerinde uygulayarak bulanık sonuçlar üretmektedir. İlk olarak, her bir giriş değerinin ne oranda hangi üyelik kümesine ait olduğu saptanmaktadır. Bu değerler kural tablosuna yerleştirilerek uygun çıkışlar elde edilmektedir. Bulanık mantık kuralları kural içerisindeki bileştiricilerin anlamlarının yorumlanması ile hesaplanmaktadır. Bu kurallar çıkarım işlemi

süresince genelleştirilmiş modus ponens yöntemi kullanmaktadır. X ve Y evrenlerinde girişi A, çıkışı B ile temsil eden iki bulanık küme tanımlansın. Bu iki bulanık küme arasındaki kural ise “Eğer A ise B” şeklinde verilsin. Bu ifadelerin oluşturduğu kural tabanı, $X \times Y$ evrenindeki bir $R_{A \rightarrow B}$ bulanık ilişkisiyle yorumlanır. Bu şekilde verilen bir bulanık ilişki; cebrik çarpım, aritmetik kural, minimum ilişki gibi değişik yöntemler kullanılarak elde edilmektedir. $R_{A \rightarrow B}$ verilen bulanık ilişki minimum ilişki yöntemiyle aşağıdaki gibi hesaplanmaktadır.

$$\mu R(x,y) = (\mu A(x) \wedge \mu B(y)) \quad (5.4)$$

“Eğer A’ ise B” şeklinde verilen bir kural, “Eğer A ise B” şeklindeki bir kurala uygulanırsa elde edilecek B’ çıkışı, A’ ile R’ nin kompozisyonu alınarak bulunmaktadır.

$$B' = A' \circ R \quad (5.5)$$

Yukarıdaki “o” işlemi bir bileşimi göstermekte olup bulanık yorumlama kuralları kullanılarak elde edilmektedir. Bu işlem, max - min yöntemiyle yorumlanırsa aşağıdaki gibi elde edilmektedir. (Klir and Folger, 1988)

$$\mu_{B'}(y) = \max [\mu_{A'}(x) \wedge \mu_R(x,y)] \quad (5.6)$$

μR ’ nin değeri “Eş. 5.6” da yerine yazılırsa ;

$$\begin{aligned} \mu_{B'}(y) &= \max [(\mu_{A'}(x) \wedge (\mu_A(x) \wedge \mu_B(y)))] \\ &= \max [\mu_{A'}(x) \wedge \mu_A(x) \wedge \mu_B] \\ \alpha &= \max (\mu_{A'}(x) \wedge \mu_A(x)) \text{ ise} \end{aligned}$$

$$\mu_{B'}(y) = \alpha \wedge \mu_B(y) \quad (5.7)$$

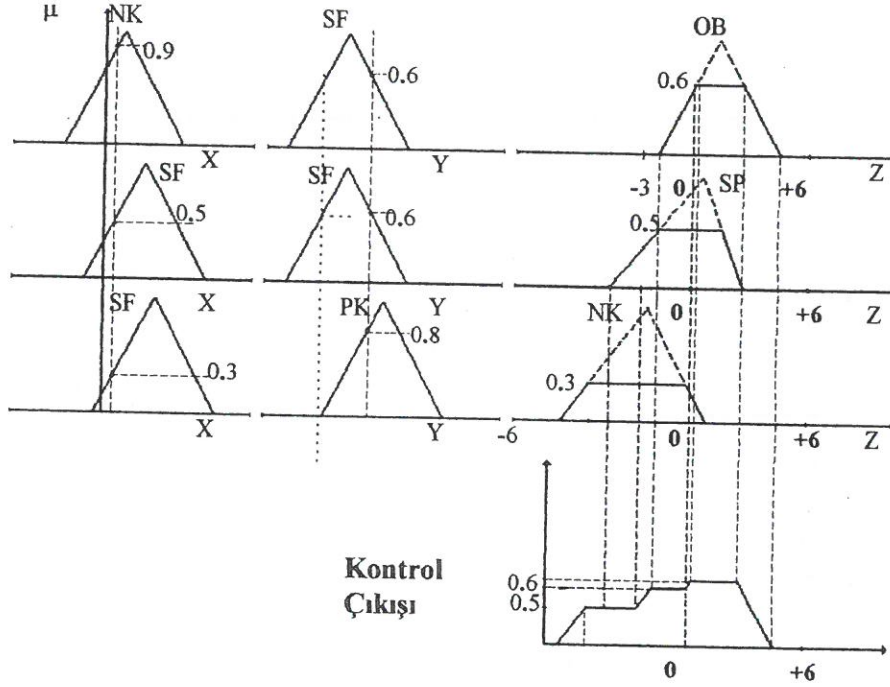
Bulanık denetleyicisi sistemlerinde n tane kural birbirine “veya” bağlacı ile bağlanır. n adet kuraldan oluşan bir kural tabanında bir çok kural aktif hale gelebilmektedir. Bu durumda minimum ilişki yöntemiyle tespit edilen kurallar, kendi arasında tekrar değerlendirmeye tabi tutulmaktadır.

Minimum ilişki yöntemi kullanılarak elde edilen çıkarım sonuçları, “veya” bağlacı kullanılarak yorumlanmaktadır. Bulanık mantıkta “veya” bağlacı max işlemine karşılık gelmektedir. İlk olarak girişler arasında minimum işlemi uygulanarak, her bir kuralın çıkış üzerinde ne kadar etkili olacağı bulunur. Sonra çıkışlar üzerinde max işlemi uygulanarak bulanık sonuç elde edilmektedir. Eğer kurallar arasında aynı çıkışı veren kurallar mevcut ise bunların en büyüğü seçilerek diğer kural iptal edilmektedir. Şekil 5.4’ de üç kural için bu işlemler gösterilmektedir. (Kıray, 1997)

Eğer A_1 ve B_1 ise, veya

Eğer A_2 ve B_2 ise, veya

Eğer A_3 ve B_3 ise, veya.



Şekil 5.4. Minimum ilişki yöntemi ile kuralların değerlendirilmesi

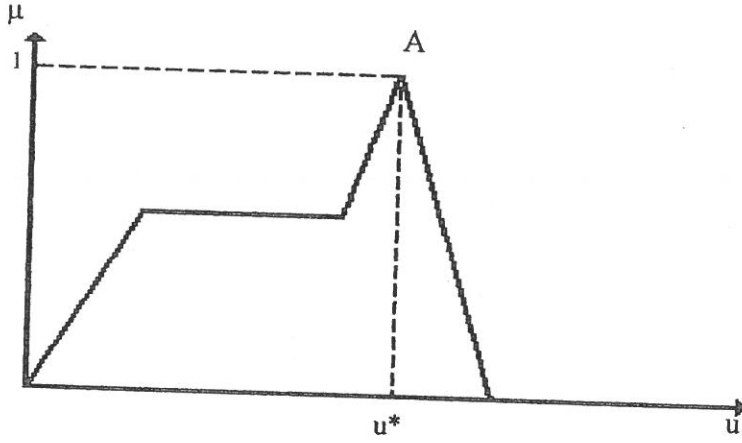
5.2.4. Berraklaştırma ünitesi

Bu ünite, çıkarım ünitesinden gönderilen kontrol işaretinin sayısal bir değer haline getirilmesini sağlamaktadır. Berraklaştırıcı, bu işlemi aşağıda kullanılan yöntemlerden birini kullanarak yapmaktadır.

1) Üyelik Fonksiyonunun Max Noktası (Max - Membership Principle):
Bu yöntemden “Yükseklik Yöntemi” olarak da söz edilmektedir. Aktif olan kuralların en büyük üyelik derecesi sayısal kontrol işareti olarak alınmaktadır. (Şekil 5.5)

$$\mu A(u^*) \geq \mu A(u)$$

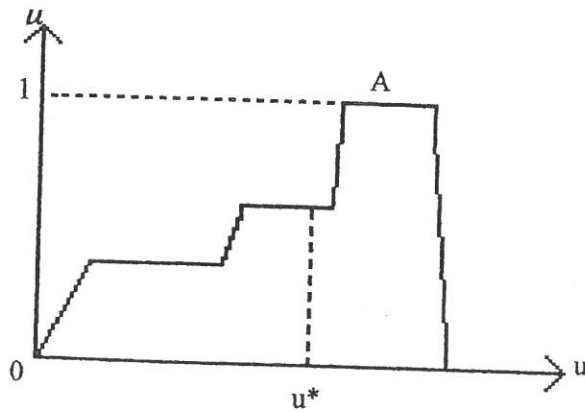
(5.8)



Şekil 5.5. Üyelik Fonksiyonlarının Max noktaları ile Berraklaştırma İşlemi

2) Merkez Yöntemi (Centroid Method): Alan merkezi ya da ağırlık merkezide denilen bu yöntem, durulama yöntemi olarak en çok kullanılan yöntemlerden biridir ve ağırlık merkezi hesaplanarak yapılmaktadır. (Şekil 5.6)

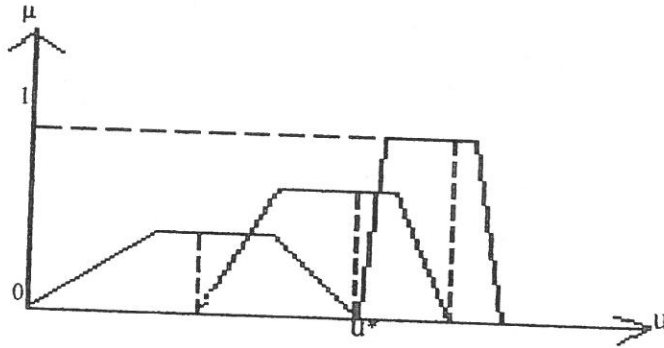
$$U^* = \frac{\int \mu(u) \cdot u \cdot dU}{\int \mu(u) \cdot dU} \quad (5.9)$$



Şekil 5.6. Merkez yöntemi ile berraklaştırma işlemi

3) Ağırlıklı Ortalama Yöntemi (Weighted Avarage Method): Bu yöntem yalnızca simetrik çıkışlı üyelik fonksiyonları için kullanılmaktadır. Her bir simetrik üyelik değerinin tepe noktası değeri belirlenerek, ortalamaların alınmasıyla yapılmaktadır. (Şekil 5.7)

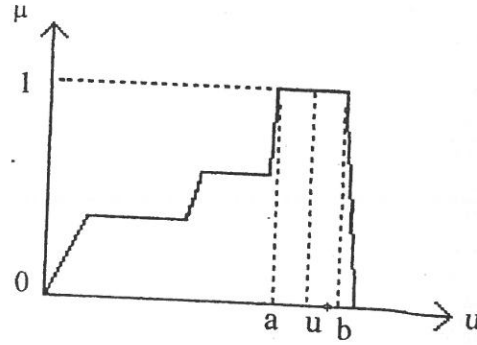
$$U^* = \frac{\sum \mu(u).u}{\sum \mu(U)} \quad (5.10)$$



Şekil 5.7. Ağırlıklı ortalama yöntemi ile berraklaştırma işlemi

4) Üyelik İşlevinin Max Noktalarının Ortalaması (Mean - Max Membership): Yüksek noktaların ortası da denilen bu yöntem, ilk yöntemle aşağı yukarı benzemektedir. Ancak üyelik işlevinin en yüksek noktası burada tek değildir. Şekil 5.8' de görüldüğü gibi bir dörtgen olabilmektedir.

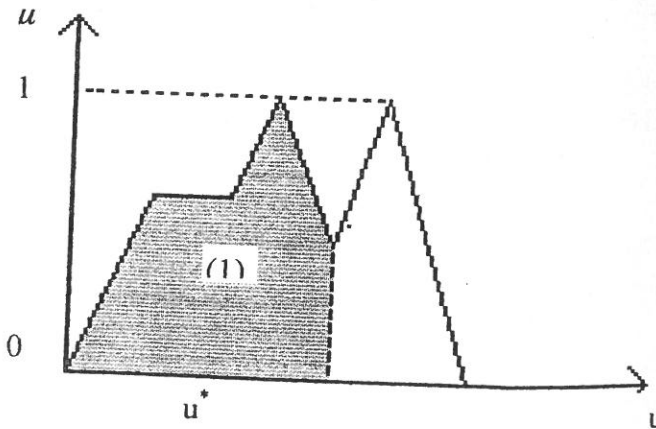
$$U^* = \frac{a + b}{2} \quad (5.11)$$



Şekil 5.8. Max noktaların ortalaması yöntemiyle berraklaştırma işlemi

5) Geniş Alan Merkezi (Center of Largest Area): Eğer bulanık çıkarımlar en az iki tane dışbükey üyelik elemanından oluşuyor ise bu yöntem kullanılabilir. Bu yöntemde dışbükey olmayan üyelik değerlerinin bileşkeleri parçalanarak A_k 'nin bir geniş parçasıdır. Zaten A_k dışbükey olsaydı, yöntem tamamıyla merkez yöntemiyle aynı olacaktı. Şekil 5.9' da görüldüğü gibi (1) numara ile belirlenmiş olan geniş alan ele alınmaktadır ki bu alan dışbükeydir.

$$U^* = \frac{\int \mu_{Am}(u) u dU}{\int \mu_{Am}(U) dU} \quad (5.12)$$



Şekil 5.9. Geniş alan merkezi metodu ile berraklaştırma işlemi

6) İlk veya son Yükselti (First or last of maxima): Bu yöntem tüm bulanık çıkışlarda uygulanabilecek bir yöntemdir. Bileşkedeki en yüksek değer $\text{hgt}(A_k) = \sup \mu_{A_k}(u)$ olmak üzere ;

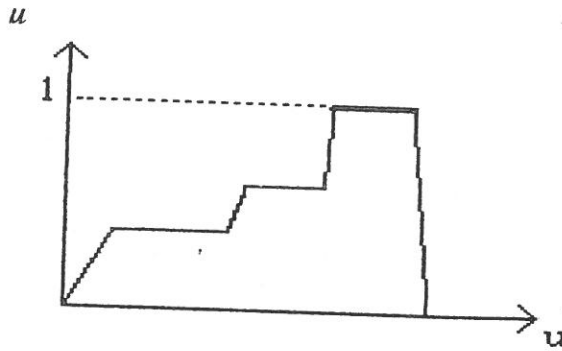
$(u \in U)$ ilk yükselti ,

$$U^* = \inf \{u \in U / \mu_{A_k}(U) = \text{hgt}(A_k)\} \quad (5.13)$$

ya da son yükselti ,

$$U^* = \sup \{u \in U / \mu_{A_k}(U) = \text{hgt}(A_k)\} \quad (5.14)$$

burada $\sup$ (supremum), en düşük yüksek sınırı ve $\inf$ (infimum) da en yüksek düşük sınırı göstermektedir. Şekil 5.10' da görüldüğü gibi ilk yükselti çözümü U_1^* veya son yükselti çözümü ise U_2^* dir.



Şekil 5.10. İlk veya son yükselti metodu ile berraklaştırma işlemi

Burada altı ayrı durulama yönteminden söz edildi. Ancak, akla "Acaba bunlardan hangisi daha iyi ?" ya da "Acaba hangisini kullanmalıyım ?" şeklinde bir soru gelmektedir. Bunun cevabı, probleme en uygun olanının seçilmesidir. Hellendorn ve Thomas 1993 yılında uygun olanın seçilmesi için beş dayanak ortaya atmıştır. Bunlar; süreklilik (continuity), belirsiz olmama (disambiguity), uygunluk (plausibility), hesapsal kolaylık (computational

simplicity) ve ağırlık yöntemidir (weighting method). Ayrıca fiziksel sistemin yapısı ve kullanıcıların deneyimleri de elbette ki durulama yönteminin seçilmesi için birer büyük dayanak noktasını oluşturmaktadır. (Günel, 1996)

6. GABDEN' NİN TASARIMI VE GERÇEKLEŞTİRİLMESİ

6.1. Giriş

Bu çalışmada, on giriş bir çıkış genel amaçlı bulanık mantık denetleyicisinin tasarımı yapılmıştır. Ayrıca bir uygulama örneği olarak verilen sistemin kontrolünde dikkate alınan işlem basamakları anlatılmıştır. GABDEN, veri sayısı bir ile on arasında değişen tüm sistemlerin denetlenmesinde kullanılabilecek niteliktedir.

GABDEN, PC kullanılarak Quick Basic 7.1 programlama dilinde hazırlanan bir programla gerçekleştirilmiştir. Bu denetleyicide bir uygulama örneği olarak, 3000W'lık ısıtıcı ile ısıtılan odanın sıcaklık kontrolü ele alınmıştır. GABDEN' de giriş parametreleri olarak iç ortam sıcaklığı, dış ortam sıcaklığı, nem ve hava kalitesi, çıkış parametresi olarak ise ısıtıcı gücü gösterilmiştir.

Bir uygulama örneği olarak seçilen bu sistemde, aşağıdaki amaçların gerçekleştirilmesi düşünülmüştür.

a) Oda soğuk ise ısıtıcı tam güçte çalıştırılarak oda sıcaklığı normal haline (18° - 19°C) getirilmesi gerekmektedir.

b) Oda çok sıcak ise ısıtıcı düşük güçte çalıştırılarak oda sıcaklığının normal haline getirilmesi gerekmektedir.

c) Odadaki nem oranı yüksek ise; ilk önce ısıtıcı tam güçte çalıştırılarak nem oranı düşürülür. Sonra sıcaklık düşürülerek oda sıcaklığı normal haline getirilmektedir.

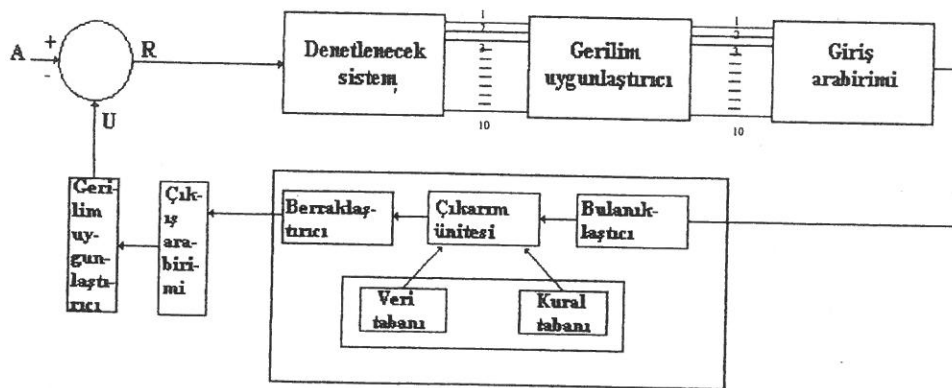
d) Hava kirliliği yüksekse, havalandırma sağlanır. Bu sırada ortamda oluşan ısı kaybı denetleyici sistem vasıtası ile telafi edilmektedir.

Gerçekleştirilen denetleyiciyle bulanık mantıkla yapılan denetleyicilerin, diğer denetleyici sistemlerle karşılaştırılması hedeflenmiştir.

6.2. GABDEN' de Kontrol Basamaklarının Oluşturulması

GABDEN' nin, denetlenecek sistemle olan bağlantısı Şekil 6.1'deki blok şemada verilmektedir.

Denetlenecek sistem çıkışından elde edilen veriler, PC'ye uygulanmadan önce uygun ara devreler kullanılarak (-9V) - (+9V) arasındaki değerlerle gösterilmesi gerekmektedir. Bu değerler, paralel arabirim devresi üzerinden PC' ye gönderilmektedir. Bulanık denetleyiciye gelen sayısal değerler, bulanıklaştırma ünitesinde bulanık değerler haline getirilmektedir. Çıkarım ünitesi, bu değerlerle kural tabanındaki değerleri karşılaştırarak berraklaştırıcı ünitesine aktarmaktadır. Bu ünite, değerler üzerinde bir ölçü değişikliği yapılarak sayısal değerler elde edilmektedir.



Şekil 6.1 Bulanık mantık denetleyicisinin sistem bağlantısı

6.2.1 Bulanıklaştırma İşlemleri

Bulanıklaştırma ünitesi, denetlenecek sistemden gelen sayısal değerler üzerinde ölçü değişikliği yaparak bulanık değerler üretmektedir. Bulanıklaştırma işlemi, üyelik fonksiyonları vasıtasıyla gerçekleştirilmektedir.

Bulanık denetleyicilerde, üyelik fonksiyonlarının seçilmesi ve gruplandırılması oldukça önemlidir. Seçilecek üyelik fonksiyonunun denetlenecek sisteme uygun olması gerekmektedir. Aksi takdirde sistemin denetlenmesi sırasında istenilen verim elde edilemez. Denetlenecek sisteme ait üyelik fonksiyonları, genelde parçalı fonksiyonlar şeklinde ifade edilmektedir.

Gerçekleştirilen denetleyicide üyelik fonksiyonları üç ayrı aralıkta tanımlanmaktadır. Tanımlanacak fonksiyon, iki veya daha az aralıkta temsil ediliyor ise diğer aralıkların boş bırakılması gerekmektedir. Bu aralıklar; $a_1 - a_2$, $a_2 - a_3$, $a_3 - a_4$ şeklinde verilsin. İki aralıkta tanımlanacak bir fonksiyon dikkate alınırsa, fonksiyon $a_1 - a_2$, $a_2 - a_3$ aralığında tanımlanarak diğer aralık boş bırakılarak geçilmelidir. Sistem, her giriş için ayrı bir üyelik fonksiyonu tanımlanacak şekilde düzenlenmiştir. Bu bize denetlenecek sistem için en uygun üyelik fonksiyonunu seçme imkanı sağlamaktadır. Denetleyici sistem, 10'dan az parametreye göre denetlenen sistemlerde kullanılırken; boş kalan girişler için üyelik fonksiyonu tanımlanmamaktadır.

Üyelik kümeleri, kümeler adlı seçenek seçilerek tanımlanmaktadır. İlk olarak her bir giriş için tanımlanacak üyelik kümesi sayısı girilmektedir. Sonra her bir giriş için tespit edilen kümeler tanımlanmaktadır. Kümeler için tespit edilen alt ve üst sınır değerleri, sabitler adlı seçenek seçilerek

girilebilir. Burada dikkat edilmesi gereken husus, alt ve üst sınır değerlerinin doğru olarak girilmesidir.

6.2.2. Kural Tabanının Oluşturulması

Bu ünite, giriş değerlerinin ikili olarak karşılaştırılması sonucunda elde edilen kontrol kuralları yer almaktadır. Bu kurallar, girişte elde edilen verilere göre sistemin nasıl davranması gerektiğini belirlemektedir.

Çizelge 6.1' de gösterildiği gibi kurallar ikili matrisler halinde düzenlenmiştir.

Çizelge 6.1. Denetleyicide Kural tabanının ikili matrisler halinde sıralanması

1.Gir.	2.Gir.	3.Gir.	4.Gir.	5.Gir.	6.Gir.	7.Gir.	8.Gir.	9.Gir.
1-2	2-3	3-4	4-5	5-6	6-7	7-8	8-9	9-10
1-3	2-4	3-5	4-6	5-7	6-8	7-9	8-10	
1-4	2-5	3-6	4-7	5-8	6-9	7-10		
1-5	2-6	3-7	4-8	5-9	6-10			
1-6	2-7	3-8	4-9	5-10				
1-7	2-8	3-9	4-10					
1-8	2-9	3-10						
1-9	2-10							
1-10								

Bu denetleyicide, tanımlanabilecek maksimum kural sayısı aşağıdaki formül ile bulunmaktadır.

$$N = K \times M \quad (6.1)$$

Burada N maksimum kural sayısını, K ikili olarak tanımlanmış matrisler arasındaki kural sayısını, M her bir giriş için oluşturulacak ikili matrislerin toplamını göstermektedir. Formül düzenlenirse;

$$N = K \times \sum_{i=1}^9 \sum_{j=1}^{10} \mu_{ij} \quad (6.2)$$

eşitliği elde edilecektir. 10 giriş ve yedi üyelik kümesi için tanımlanabilecek max kural sayısı;

$$\begin{aligned} N &= (7 \times 7) \times \sum_{i=1}^9 \sum_{j=1}^{10} \mu_{ij} \\ N &= 49 \times (9+8+7+6+5+4+3+2+1) \\ N &= 2205 \end{aligned} \quad (6.3)$$

olarak bulunacaktır. Çizelge 6.2' de üç giriş ve her giriş için iki üyelik kümesi tanımlanmış bir sistemin kural tabanına örnek gösterilmektedir.

Çizelge 6.2. Üç girişli bir sistem için oluşturulan kural tabanı

		2. Giriş				3. Giriş			
		S0		SI		Y		D	
1. Giriş						1. Giriş			
B		B	0			B	B		
K		0	K			0	K		
								2. Giriş	
								S0	
								SI	
								B	B
								K	0

6.2.3. Çıkarım İşlemleri

Denetleyici sistemde, çıkarım işlemleri çözüm 12 başlığı altında verilen programda yapılmaktadır. Burada yapılan işlemler, aşağıdaki gibi sıralanabilir.

1) Girişten gelen bilgiler, paralel arabirim devresiyle 12 bit olarak okutulur.

2) Okutulan port bilgilerinin hangi kümenin elemanı olduğu tespit edilir. Sonra ilgili kümenin üyelik fonksiyonu seçilerek, giriş değerleri bulanıklaştırılır.

3) Bulanıklaştırıcıdan gelen bulanık değerlere göre, aktif olan kurallar işaretlenir ve her bir ikili karşılaştırma için tanımlanmış çıkış kazancı tespit edilir.

4) Üyelik ağırlıklarının hesaplanması için max - min metodu kullanılmıştır. Bu metotta, önce üyelik ağırlıkları arasında minimum işlemi gerçekleştirilir. Daha sonra aynı çıkış değerine sahip olan aktif kurallar içerisinde üyelik ağırlığı daha büyük olan kural seçilerek sonuca ulaşılır.

6.2.4. Berraklaştırma İşlemleri

Bu bölümde, çıkarım ünitesinden elde edilen bulanık değerler üzerinde ölçek değişikliği yapılarak, kontrol edilecek sistem için gerekli olan sayısal değerler elde edilmektedir. Denetleyici sisteminde, bu işlem için ağırlık merkezi yöntemi kullanılmaktadır.

$$U = \frac{\sum_{i=1}^n W_i \cdot \mu(w_i)}{\sum_{i=1}^n \mu(w_i)} \quad (6.4)$$

Burada n kural sayısını, W_i kontrol çıkışını ve $\mu(w_i)$ üyelik ağırlıklarını göstermektedir. Elde edilen sayısal değer arabirim devresiyle denetlenecek sisteme uygulanmaktadır.

6. 3. Bilgisayara Giriş Çıkış Sistemi

Denetleyici sistem, bir PC makine ve bulanık kontrol mantığı kullanılarak yapılmıştır. Algılayıcılardan gelen bilgilerin bilgisayarla iletişime geçebilmesi için bir giriş-çıkış arabirim devresi kullanılmıştır. Bu işlem için Decision-Computer International Co. Ltd . firmasının üretimi olan 12 Bit analog-sayısal (A/D) ve sayısal-analog (D/A) dönüştürücü kullanılmıştır.

6.3.1. 12 Bit A/D-D/A Giriş Çıkış Kartı

12 Bit A/D - D/A Giriş Çıkış Kartı, PC/XT, PC/AT veya uyumlu bilgisayarlar için yüksek doğrulukta bir veri dönüştürme sistemidir. 1 adet 12 bit sayısal - analog kanal ve 16 adet 12 bit analog - sayısal kanal içermektedir. D tipi 25 bacaklı konnektörü vardır.

6.3.1.1 A/D - D/A Giriş - Çıkış Kartının Özellikleri

A/D-D/A Kartın belirleyici özellikleri Çizelge 6.1'deki gibidir. Giriş-Çıkış port adresleri JP1 (1 Nolu kısa devre tırnağı) ile seçilebilen &H278-27F veya &H2F8-2FF adresleri vardır. Kartın (0) - (9) veya (-9) - (9) volt arasında bilgi alış verişini JP2 ile seçebiliriz. 2 nolu kısa devre tırnağı seçildiği zaman (0) - (9) volt arasını, bir nolu kısa devre tırnağı seçildiğinde ise (-9) - (9) volt arasını seçmiş olacağız. A/D dönüştürücü için çift kutup (-9 - 9 V) veya tek kutup seçme işlemi JP3 ile seçilmektedir. 3 nolu kısa devre tırnağı seçilerek çift kutup çalışma şekli gerçekleştirilir.

Çizelge 6.3 A/D - D/A Kartın Belirleyici Özellikleri

D/A Bölümü	A/D Bölümü
1 Adet 12 bit kanalı destekler	16 Adet 12 bit kanalı destekler
Ayarlanabilir giriş voltajı: Tek kutup: (0)- (9) volt. Çift kutup: (-9)- (9) volt.	Ayarlanabilir giriş voltajı: Tek kutup: (0)- (9) volt. Çift kutup: (-9)- (9) volt.
Tek kutup veya çift kutup olması JP2 ile seçilebilir.	Tek kutup veya çift kutup olması JP3 ile seçilir.
Akım ayar zamanı 500 ns.	Dönüştürme zamanı her kanal için 60 μ sn

6.3.1.2 Donanım Düzeni

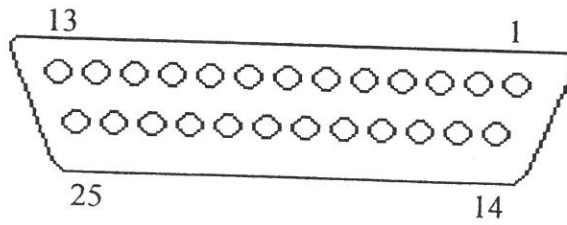
A/D - D/A kart için uygun ayarlar aralıkları aşağıda verilmiştir. Giriş - Çıkış port adresleri, &H278 - 27F veya &2F8 - 2FF olarak seçilebilmektedir. Bu adreslerin ayrıntılı karşılıkları Çizelge 6.4' te verilmiştir.

Çizelge 6.4 A/D - D/A Kartın Adreslerinin Ayrıntılı Karşılıkları.

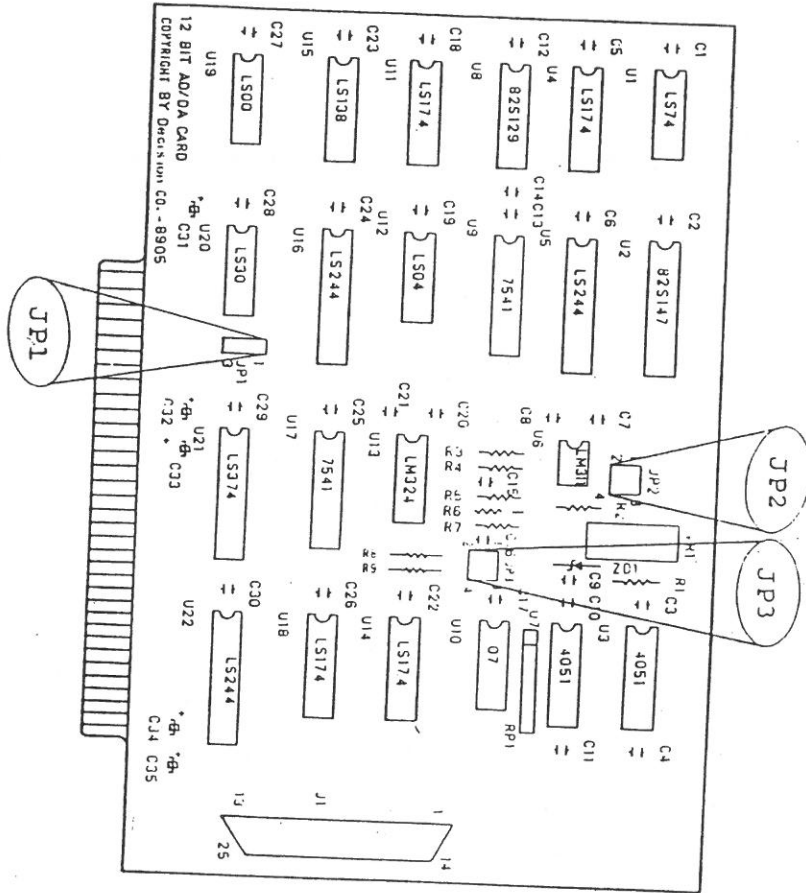
&H278	&H2F8	Çıkış A/D Kanal Numarası. (Alçak kısım)
279	2F9	Giriş A/D düşük bayt veri. (8 Bit)
27A	2FA	Giriş A/D yüksek bayt veri. (Alçak kısım)
27B	2FB	Temizle A/D register.
27C	2FC	A/D Dönüşüm döngüsü. (Alçak)
27D	2FD	A/D Dönüşüm döngüsü. (Yüksek)
27E	2FE	Çıkış D/A düşük byte veri. (8 Bit)
27F	2FF	Çıkış D/A yüksek byte veri.(Alçak kısım)

Çizelge 6.5 D Tipi Konnektör Bacak Sırası ve Fonksiyonları

Bacak No	Fonksiyonu	Bacak No	Fonksiyonu
1	TOPRAK	14	TOPRAK
2	D/A ÇIKIŞ	15	-5 V
3	+5 V	16	TOPRAK
4	TOPRAK	17	A/D - 15. KANAL
5	A/D - 0. KANAL	18	A/D - 14. KANAL
6	A/D - 1. KANAL	19	A/D - 13. KANAL
7	A/D - 2. KANAL	20	A/D - 12. KANAL
8	A/D - 3. KANAL	21	A/D - 11. KANAL
9	A/D - 4. KANAL	22	A/D - 10. KANAL
10	A/D - 5. KANAL	23	A/D - 9. KANAL
11	A/D - 6. KANAL	24	A/D - 8. KANAL
12	A/D - 7. KANAL	25	-12 V
13	+12 V		



Şekil 6.2 A/D-D/A Karta Ait 25 Bacaklı Konnektörün Bacak Dizilişi



Şekil 6.3 A/D - D/A Kartın Yerleşim Düzeni

6.3.1.3 Kısa Devre Bacaklarının Ayarları

Kısa devre tınakları JP1, JP2 ve JP3 olarak Şekil 6.2' de görülmektedir. JP1' de, 1 ve 2 nolu bacaklar kısa devre yapılırsa, &H278-27F adresleri seçilmiş olmaktadır. 2 ve 3 nolu bacaklar kısa devre yapılırsa, &H2F82FF adresleri seçilmiş olur. Bilgisayar sistemine uygunluk için 1 ve 2 nolu bacakların seçilmesi uygundur.

JP2' de, 3 ve 4 nolu bacaklar kısa devre yapılırsa, çift kutup D/A çıkış elde edilir. 1 ve 2 nolu bacaklar kısa devre yapılırsa, tek kutup D/A çıkış elde edilir.

JP3' de, 3 ve 4 nolu bacaklar kısa devre yapılırsa, çift kutup A/D giriş elde edilir. 1 ve 2 nolu bacaklar kısa devre yapılırsa, tek kutup A/D giriş elde edilir.

6.3.1.4. Programlama ve Test Teknikleri

MS/DOS altında programlama işlem sırası:

Port = 760 (&H2F8 - 2FF seçildiği zaman)

Port = 632 (&H278 - 27F seçildiği zaman)

1. Analog - sayısal (A/D) işlemi için işlem sırası:

a. Çıkış portu için çıkış kanal numarasını kanal olarak tanımlayı gerçekleştiriniz

OUT port,channel

b. Kaydediciyi belirlemek için (clear register):

OUT (Port + 3), 0

c. Dönüştürmeye başla

FOR I=1'den 5'e

A=INP(port + 4)

NEXT I

FOR I=1'den 9'a

A= INP(port + 5)

NEXT I

d. Yüksek bayt'ı oku (alçak kısım)

$$C = \text{INP}(\text{port} + 2)$$

$$\text{HB} = [C/16 - \text{INT}(C/16)] \cdot 16$$

e. Düşük bayt'ı oku (8 bit)

$$\text{LB} = \text{INP}(\text{port} + 1)$$

f. Veri :

$$\text{A/D} = \text{HB} \cdot 256 + \text{LB}$$

2. Sayısal - Analog (A/D) dönüştürme için işlem sırası :

a. Yüksek bayt'ı çıkışa gönder (alçak kısım)

$$\text{OUT port} + 7, \text{Hdata}$$

b. Düşük bayt'ı çıkışa gönder (8 bit)

$$\text{OUT port} + 7, \text{Ldata}$$

6.4. Uygulama Örneği

Bu uygulama örneğinde, dört giriş, tek çıkış bir ısıtıcı sisteminin bulanık mantıkla denetlenmesi ele alınmıştır. Bulanık denetleyicisi için giriş parametreleri olarak; iç ortam sıcaklığı, dış ortam sıcaklığı, nem ve hava kirliliği, çıkış parametresi olarak da ısıtıcı gücü ele alınmıştır.

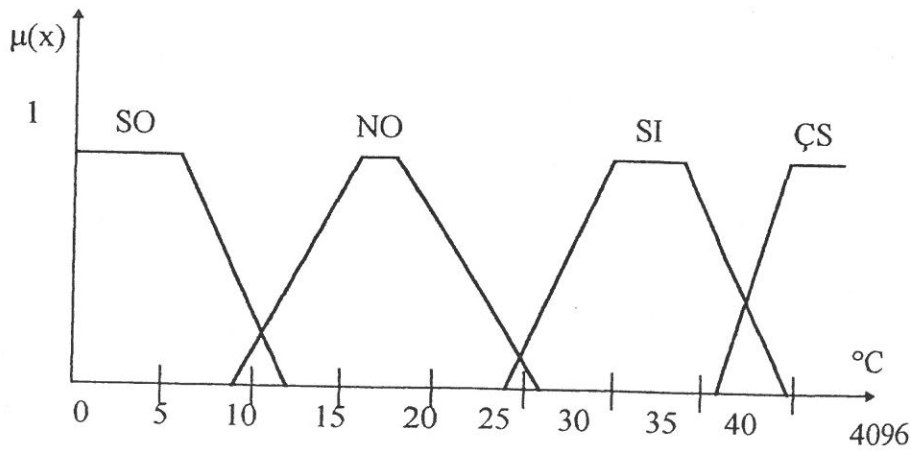
6.4.1. Bulanık Kontrol Basamaklarının Oluşturulması

İç sıcaklık, dış sıcaklık, nem ve hava kalite sensörlerinden alınan Analog bilgiler minimum -9V ile maximum +9V arasında uygun devreler kullanılarak paralel arabirim devresine uygulanır. Arabirim devresi bu verileri sayısal değere dönüştürerek PC' ye uygular. Bundan sonra program işletilerek Bulanık kontrol basamakları uygulanmaya başlar.

6.4.1.1. Bulanıklaştırma İşlemleri

Bulanıklaştırma bölümünde iç sıcaklık, dış sıcaklık, nem ve hava kalite algılayıcılarından alınan bilgiler bulanık forma dönüştürülmektedir. İlk olarak her bir giriş değerini, dilsel değişkenler haline getiren üyelik fonksiyonlarının oluşturulması gerekmektedir. Bunun için seçilebilecek üyelik kümeleri ve her bir girişe ait alt-üst sınır değerleri belirlenmektedir. İç sıcaklık için soğuk, normal, sıcak ve çok sıcak olmak üzere dört ayrı üyelik kümesi; dış sıcaklık için çok soğuk, soğuk, az soğuk, serin, normal ve sıcak olmak üzere 6 ayrı üyelik kümesi; nem basıncı için yüksek ve düşük olmak üzere iki ayrı üyelik kümesi; hava kalitesi için kirli ve temiz olmak üzere iki ayrı üyelik kümesi tanımlanmaktadır.

İç sıcaklığa ait yamuk bölgeler oluşturacak şekilde seçilmiştir. Bunun sebebi; sistemin sürekli normal sıcaklığı yakalamaya çalışmasından dolayıdır. Normal bölgesi dar tutularak kısa sürede sistemin ortam ısını yakalaması hedeflenmiştir. İç ortam sıcaklığına ait üyelik fonksiyonlarının sınırları Çizelge 6.6' daki gibi tayin edilmiştir.

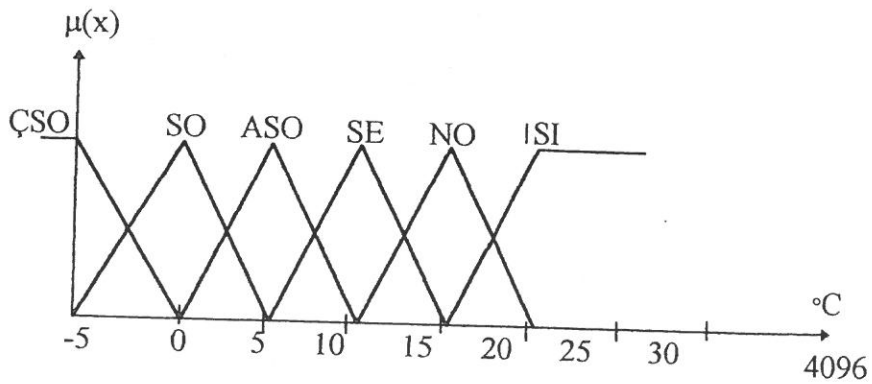


Şekil 6.4. İç sıcaklık bilgisine ait üyelik fonksiyonları

Çizelge 6.6. İç ortam sıcaklığına ait üyelik fonksiyonlarının oluşturulması

Üyelik Kümeleri	Fonksiyon	Aralık Değeri
SOĞUK (SO)	$\mu(x)=1$	$-\infty \leq x \leq 7$
	$\mu(x) = (12-x)/(12-7)$	$7 \leq x \leq 12$
NORMAL (NO)	$\mu(x) = (10-x)/(10-18)$	$10 \leq x \leq 18$
	$\mu(x)=1$	$18 \leq x \leq 19$
	$\mu(x) = (27-x)/(27-19)$	$19 \leq x \leq 27$
SICAK (SI)	$\mu(x) = (24-x)/(24-29)$	$24 \leq x \leq 29$
	$\mu(x)=1$	$29 \leq x \leq 32$
	$\mu(x) = (37-x)/(37-32)$	$32 \leq x \leq 37$
ÇOK SICAK (ÇS)	$\mu(x) = (34-x)/(34-39)$	$34 \leq x \leq 39$
	$\mu(x)=1$	$39 \leq x \leq +\infty$

Dış sıcaklık bilgisine ait üyelik fonksiyonları üçgensel tipte seçilmiştir. Bu bilgiye ait üyelik fonksiyonları şekil 6.5' de gösterilmiştir. İç ortam sıcaklığının 19°C' de sabit tutulabilmesi için, dış ortamdaki bilgilerin hassas bir şekilde değerlendirilmesi gerekir. Üyelik fonksiyonlarının sınırları Çizelge 6.7' de olduğu gibi tayin edilmiştir.

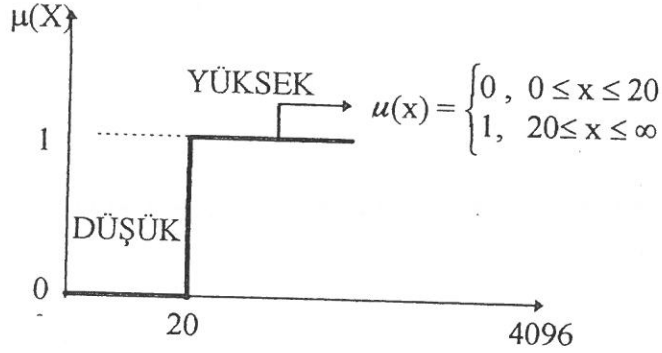


Şekil 6.5. Dış sıcaklık bilgisine ait üyelik fonksiyonları

Çizelge 6.7. Dış ortam sıcaklığına ait üyelik fonksiyonlarının oluşturulması

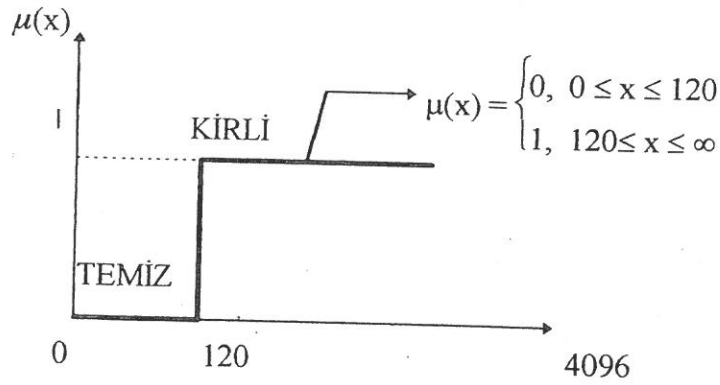
Üyelik Kümeleri	Fonksiyon	Aralık değerleri
ÇOK SOĞUK (ÇSO)	$\mu(x)=1$ $\mu(x)=(0-x)/(0+5)$	$-\infty \leq x \leq -5$ $-5 \leq x \leq 0$
SOĞUK (SO)	$\mu(x)=(-4-x)/(-4-1)$ $\mu(x)=(6-x)/(6-1)$	$-4 \leq x \leq 1$ $1 \leq x \leq 6$
AZ SOĞUK (ASO)	$\mu(x)=(1-x)/(1-6)$ $\mu(x)=(11-x)/(11-6)$	$1 \leq x \leq 6$ $6 \leq x \leq 11$
SERİN (SE)	$\mu(x)=(6-x)/(11-6)$ $\mu(x)=(16-x)/(16-11)$	$6 \leq x \leq 11$ $11 \leq x \leq 16$
NORMAL (NO)	$\mu(x)=(11-x)/(11-16)$ $\mu(x)=(21-x)/(21-16)$	$11 \leq x \leq 16$ $16 \leq x \leq 21$
SICAK (SI)	$\mu(x)=(16-x)/(16-21)$ $\mu(x)=(26-x)/(26-21)$	$16 \leq x \leq 21$ $21 \leq x \leq 26$

Nem bilgisi, sıcaklık bilgisinde farklı bir büyüklük olduğu için, nemin yüksek ve düşük olması dikkate alınarak iki ayrı bulanık küme olarak tanımlanmıştır. Bunun sebebi oda içerisindeki nem oranının fazla yüksek olmamasından dolayıdır. Bu bulanık kümeler düşük (D) ve yüksek (Y) olarak isimlendirilmiştir. Üyelik fonksiyonları kare şeklinde seçilmiştir. Fonksiyon 0 - 20 aralığında sıfır, 20 - ∞ aralığında 1 değerini almaktadır. Şekil 6.6' da nem bilgisine ait üyelik fonksiyonu gösterilmiştir.



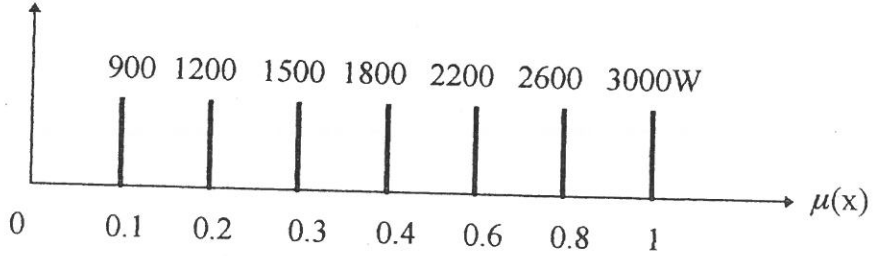
Şekil 6.6. Nem bilgisine ait üyelik fonksiyonu

Hava kalite bilgisine ait üyelik fonksiyonu da, nem bilgisine benzer şekilde tanımlanmıştır. 0 - 120 aralığında fonksiyon 1 değerini alacaktır. Şekil 6.7' de hava kalite bilgisine ait üyelik fonksiyonu gösterilmiştir.



Şekil 6.7. Hava kalite bilgisine ait üyelik fonksiyonu

Çıkış üyelik fonksiyonları, singleton fonksiyon olarak tanımlanmış ve 7 ayrı üyelik kümesi ile ifade edilmiştir. Şekil 6.8'de çıkış bilgisine ait üyelik fonksiyonu gösterilmiştir.



Şekil 6.8. Çıkışa ait üyelik fonksiyonları.

6.4.1.2. Kural Tabanının Oluşturulması

Kural tabanı, dört giriş değerine göre ikili matrisler halinde tanımlanmıştır ve üyelik fonksiyonları ile elde edilmiş bölgelere göre düzenlenmiştir. denetlenecek sistem için tanımlanan Kural tabanı Çizelge 6.8' te gösterilmiştir.

Çizelge 6.8. Kural tabanının oluşturulması

1. Giriş İç Sıcaklık	2. Giriş Dış Sıcaklık					
	ÇSO	SO	ASO	SE	NO	SI
SO	1	1	0.8	0.6	0.2	0.1
NO	0	0	0	0	0	0
SI	0	0	0	0	0	0
CS	0	0	0	0	0	0

(a)

1. Giriş İç Sıcaklık	3. Giriş Giriş Nem Basıncı	
	D	Y
SO	0.8	1
NO	0	1
SI	0	0.6
CS	0	0.2

(b)

Dış sıcaklık:

$$\mu_{NO}(15) = (11-15)/(11-16) \Rightarrow \mu_{NO}(15) = 0.8$$

$$\mu_{SE}(15) = (16-15)/(16-11) \Rightarrow \mu_{SE}(15) = 0.2$$

Nem basıncı:

$$\mu_D(17) = 0$$

Hava kalitesi:

$$\mu_{KI}(270) = 1$$

Bu değerlere göre aktif olan kurallar ve minimum değerleri şöyle bulunmuştur.

	İç Sı. Girişi	Dış Sı. Girişi	Isıtıcı
1	0.2 (SO)	0.8 (NO)	0.2

$$W_1 = \min(0.2; 0.8) = 0.2$$

	İç Sı. Girişi	Dış Sı. Girişi	Isıtıcı
2	0.2 (SO)	0.2 (SE)	0.6

$$W_2 = \min(0.2; 0.2) = 0.2$$

	İç Sı. Girişi	Dış Sı. Girişi	Isıtıcı
3	0.125 (NO)	0.8 (NO)	0

$$W_3 = \min(0.125; 0.8) = 0.125$$

	İç Sı. Girişi	Dış Sı. Girişi	Isıtıcı
4	0.125 (NO)	0.2 (SE)	0

$$W_4 = \min(0.125; 0.2) = 0.125$$

	İç Sı. Girişi	Hava Kalitesi	Isıtıcı
5	0.2 (SO)	1 (Ki)	0.8

$$W_5=(0.2;1)=0.2$$

	İç Sı. Girişi	Hava Kalitesi	Isıtıcı
6	0.125 (NO)	1 (Ki)	0.4

$$W_6=(0.125;1)=0.125$$

	İç Sı. Girişi	Nem Basıncı	Isıtıcı
7	0.2 (SO)	0 (D)	0.8

$$W_7=\min(0.2;0)=0$$

	İç Sı. Girişi	Nem Basıncı	Isıtıcı
8	0.125 (NO)	0 (D)	0

$$W_8=\min(0.125;0)=0$$

	Dış Sı. Girişi	Nem Basıncı	Isıtıcı
9	0.8 (NO)	0 (D)	0

$$W_9=\min(0.8;0)=0$$

	Dış Sı. Girişi	Hava Kalitesi	Isıtıcı
10	0.8 (NO)	1 (Ki)	0

$$W_{10}=\min(0.8;1)=0.8$$

	Dış Sı. Girişi	Nem Derecesi	Isıtıcı
11	0.2 (SE)	0 (O)	0

$$W_{11}=\min(0.2;0)=0$$

	Dış Sı. Girişi	Hava Kalitesi	Isıtıcı
12	0.2 (SE)	1 (Ki)	0.1

$$W_{12}=\min(0.2;1)=1$$

	Nem Basıncı	Hava kalitesi	Isıtıcı
13	0 (D)	1 (K)	0.1

$$W_{13}=\min(0;1)=0$$

Buna göre; kuralların kontrol çıkışı üzerindeki etkileri aşağıdaki gibi yorumlanabilir.

- 1. Kural, ısıtıcı çıkışının 0.2 nolu kademesine 0.2 oranında etkili,
- 2. Kural, ısıtıcı çıkışının 0.6 nolu kademesine 0.2 oranında etkili,
- 3. Kural, ısıtıcının 0 nolu kademesine 0.125 oranında etkili,
- 4. Kural, ısıtıcının 0 nolu kademesine 0.125 oranında etkili,
- 5. Kural, ısıtıcının 0.8 nolu kademesine 0.2 oranında etkili,
- 6. Kural, ısıtıcının 0.4 nolu kademesine 0.125 oranında etkili,
- 7. Kural, ısıtıcının 0.8 nolu kademesine 0 oranında etkili,
- 8. Kural, ısıtıcının 0 nolu kademesine 0 oranında etkili,
- 9. Kural, ısıtıcının 0 nolu kademesine 0 oranında etkili,
- 10. Kural, ısıtıcının 0 nolu kademesine 0.8 oranında etkili,
- 11. Kural, ısıtıcının 0 nolu kademesine 0 oranında etkili,
- 12. Kural, ısıtıcının 0.1 nolu kademesine 0.2 oranında etkili,
- 13. Kural, ısıtıcının 0.1 nolu kademesine 0 oranında etkilidir

Çıkışa ait üyelik ağırlıklarının watt cinsinden değeri aşağıda verilmiştir.

0.1	0.2	0.3	0.4	0.6	0.8	1
900	1200	1500	1800	2200	2600	3000

Bu çalışmada kontrol çıkışının değeri hesaplanırken ağırlık merkezi yöntemi kullanılmıştır. Buna göre çıkış değeri aşağıdaki gibi elde edilmiştir. Buna göre çıkış değeri aşağıdaki gibi elde edilmiştir.

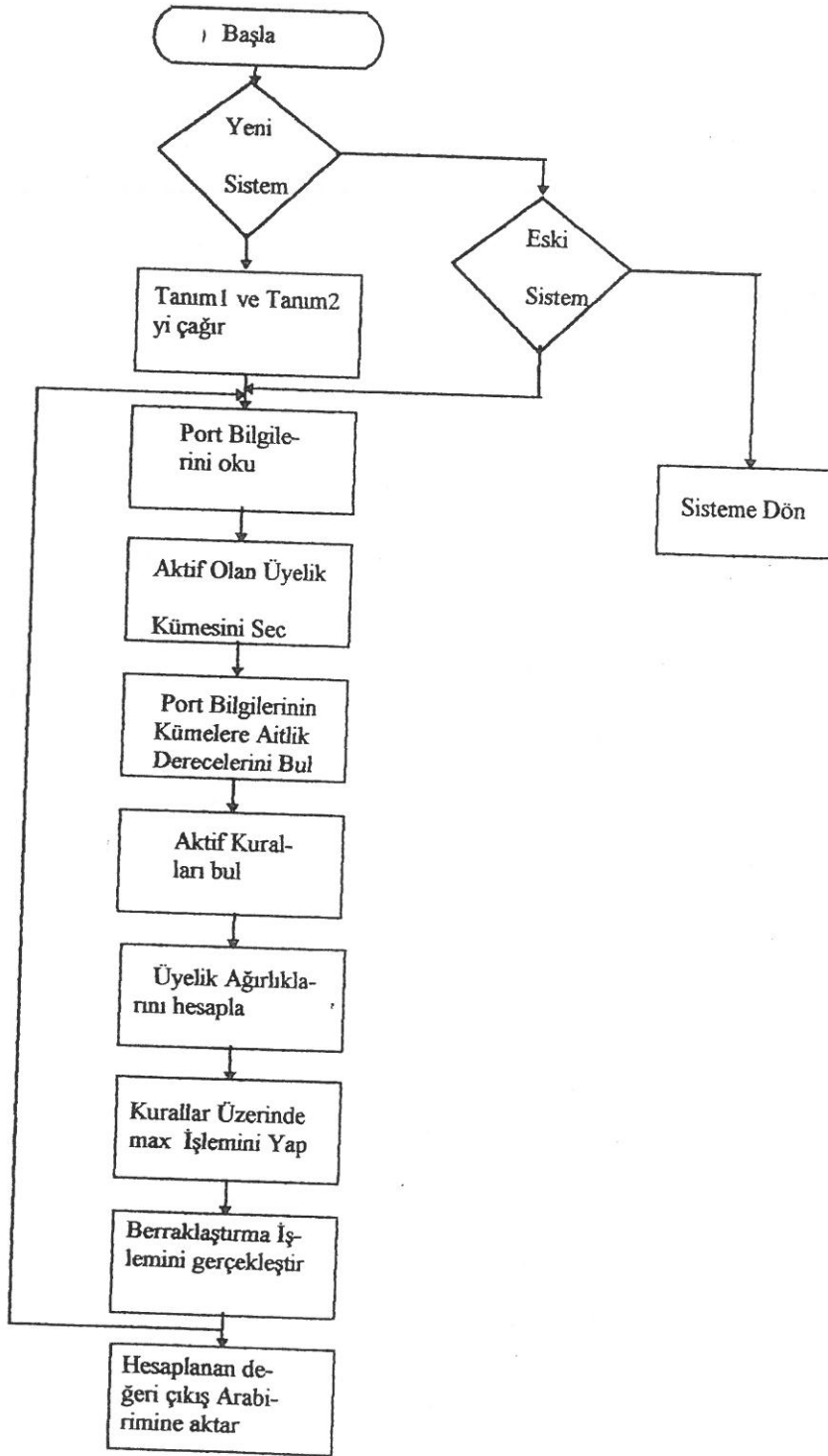
$$\begin{aligned} \text{Isıtıcı kont. çıkışı} = & (0.2 \times 1200 + 0.2 \times 2200 + 0.125 \times 0 + 0.125 \times 0 + 0.2 \times 2600 + \\ & 0.125 \times 1800 + 0.8 \times 0 + 0 \times 0 + 0 \times 0 + 0.8 \times 0 + 0 \times 0 + 0.2 \times 900 + 0 \times 900) \\ & (0.2 + 0.2 + 0.125 + 0.125 + 0.2 + 0.125 + 0 + 0 + 0 + 0.8 + 0 + 0.2 + 0) \end{aligned}$$

$$U = 812.65$$

$$\%U = \frac{812.65}{3000} = \% 27$$

Bu sonuç, ısıtıcının tam gücünün %27' si oranında çalıştırılacağını göstermektedir. Bu durum, çıkışa bağlanan bir voltmetre ile gözlenmiştir.

Şekil 6.9' da denetleyici sisteme ait ana programın akış şeması verilmiştir.



Şekil 6.9. Denetleyici sisteme ait programın ana akış şeması

7. SONUÇ VE ÖNERİLER

Genel amaçlı bir bulanık mantık denetleyicisi olarak gerçekleştirilen bu çalışmada, bulanık mantık denetleyicilerinin klasik mantık denetleyicilerine göre daha hassas bir denetim yaptığı gözlenmiştir. Özellikle lineer olmayan sistemlerin denetlenmesinde bulanık mantık denetleyicilerinin kullanılması gerçeği ortaya çıkmıştır. Bunun sonucunda, denetlenecek sistem üzerinde az bir kural kullanılarak sistemin daha hassas bir şekilde denetlenmesi mümkün olacaktır.

Bu çalışmada, bir uygulama örneği olarak bir odanın sıcaklık kontrolü yapılmıştır. Maliyetin düşük olması için denetlenecek sisteme gönderilecek verinin değişimi, bir voltmetre aracılığı ile gözlenmiştir. Bu sistemde, ilk anda oda çok soğuk veya nemli ise ısıtıcı tam güçte çalıştırılmaktadır. Bu durum PC' den gönderilen değerin maksimum bir değer alması ile gözlenmiştir. Odada hava çok kirli ise pencereler açık tutularak ortamın ısısı sabit tutulacak şekilde PC' den gerekli değer gönderilmiştir. Bu işlemler kural tabanındaki kuralların denetlenen sistem için uygun şekilde tanımlanmasıyla gerçekleştirilmiştir.

GABDEN' de on girişin bulunması denetlenecek sisteme ait verilerin tamamının kullanılmasına olanak sağlamaktadır. Bu durum, denetlenecek sistemin daha hassas bir şekilde kontrol edilmesini sağlayacaktır. GABDEN' nin tasarımında PC kullanılması GABDEN' nin esnek bir yapı teşkil etmesine sebep olmuştur. Bundan dolayı GABDEN, deneysel amaçlı bir bulanık mantık denetleyicisi olarak kullanılabilme özelliğine sahiptir. Denetlenecek sistem bilgileri, GABDEN' e bilgilerin klavyeden girilmesi ile tanımlanmıştır. Sistemde çıkarım ünitesinde sadece minimum ilişki metodunun kullanılması, berraklaştırma ünitesinde de ağırlık merkezi metodunun kullanılması

kısıtlayıcı bir özellik olarak karşımıza çıkmaktadır. Bunun giderilmesi için çıkarım ünitesine sınırlı çarpım, kesin çarpım, aritmetik kural, maksimum kural gibi bulanık ilişki yöntemlerinin ve berraklaştırma ünitesine de maksimum nokta, merkez yöntemi, ağırlıklı ortalama yöntemi, maksimum noktaların ortalaması yöntemi gibi berraklaştırma yöntemlerinin eklenmesi gerekmektedir.

KAYNAKLAR

- Bağcıoğulları F., 1995; Bağcıoğulları C.H., 1995 "Nesne Tanımlamada Yeni Bir Yaklaşım : Bulanık Kümeler" Tübitak M.A.M.; Kocaeli
- Baba F., 1995 "Triga Mark-II Reaktörünün Bulanık Kontrolü" İ.T.Ü. Doktora Tezi
- Başbuğ A., 1994 "Bulanık Teknoloji" Byte Şubat 94
- Çiftçi T., 1996 "Bulanık Kontrol Sistemlerinin Önermeler Mantığı ile Modellenmesi" Otomatik Kontrol Ulusal Toplantısı TOK 96
- Doğueri A.K., 1996 "Bulanık Mantık ve Bulanık Denetleyiciler" Otomasyon Şubat 96
- Günel Ü., 1997 "Bulanık Mantık" Otomasyon 1997
- Kaynak M., 1992 "Süreç Denetiminde Bulanık Mantık" Otomasyon Temmuz-Ağustos 1992
- Kıray V., 1997 "Oda İçi Sıcaklık ve Hava Kalitesinin Mikrodenetleyici ve Bulanık Mantıkla Kontrolü" G.Ü. Yüksek Lisans Tezi
- ✓ Kichert W.J.M., 1978 "Fuzzy Theories an Decision Making" Martinus Nijhoff Social Scienses Devision, London
- ✓ Klir G.J., 1988; Folger T.A., 1988 "Fuzzy Sets, Uncertainty and

Information" University of NewYork

√ Pedryez W., 1989 "Fuzzy Control and Fuzzy Systems" Reserch Studies Press
LTD Taunton, England

Taşkın H.,1996; Gümüştas R., 1996 "Trafik Akış Sistemlerinde Bulanık
Denetim" Otomatik Kontrol Ulusal Toplantısı TOK 96

√Zadeh L.A., 1987 "Fuzzy Set and Applications" Wiley, Interscience
Publication, Newyork

EKLER

- EK-1 : DENETLEYİCİ SİSTEME AİT QUICK BASIC 7.1 PROGRAMI
EK-2 : YÜKSEK HASSASİYETLİ DERECE CİNSİNDEN ISI ÖLÇERLER

EK-1: ANA PROGRAM

"Bu bölümde; tanımlanacak sisteme yada tanımlanmış sisteme ait işlemlerin seçimi yer almaktadır."

```

DECLARE SUB SİL ()
DECLARE SUB GONE ()
DECLARE SUB İCONMAKE ()
DECLARE SUB ALTMENUBUILD
(ALTMENU$, KOD$)
DECLARE SUB CLOSEMOUSE () DEC-
LARE SUB İNSTALL ()
DECLARE SUB UNİNSTALL () DECLARE
SUB YAZSKANT ()
DECLARE SUB SİLME () DECLARE SUB
MESAJ ()
DECLARE SUB TANİMALT1 () DECLARE
SUB SECMENUO ()
DECLARE SUB STDMENU () DECLARE
SUB YARDIM ()
DECLARE SUB GİMENU () DECLARE
SUB EKTRAN2 ()
DECLARE SUB EKTRAN1 () DECLARE
SUB SEC3 ()
DECLARE SUB TUS () DECLARE SUB
TANİMALT ()
DECLARE SUB ALMENU1 (ÇIKİS) DEC-
LARE SUB ALMENU2 (ÇIKİS)
DECLARE SUB ALMENU3 (ÇIKİS) DEC-
LARE SUB MENUİNSTALL (K1, K2)
DECLARE SUB SECONDİTEM () DEC-
LARE SUB KONTROL1 (KJ)
DECLARE SUB KONTROLANA () DEC-
LARE SUB BULANIKYAZ ()
DECLARE SUB YAZI (C) DECLARE SUB
QWAIT ()
DECLARE SUB FRAMEİCON (X1, X2, Y1,
Y2, BACKGROUND)
DECLARE SUB SECMENU () DECLARE
SUB FİRSTMENU ()
DECLARE SUB FRAME (ALT, UST, SAG,
SOL)
DECLARE SUB TANİM ()
COMMON SHARED KATAR$, ESC$,
BACKSPACES$, İNSETS$,
COMMON SHARED UP$, DOWN$,
LLEFT$, RRIGHT$, SPCE$, ENTER$,
F1$:COMMON SHARED F2$, F3$, F4$,
F5$, F6$, F7$, F8$, F9$, F10R$
COMMON SHARED SAYI, X, Y, S, L
A$ = CHR$(223)
STACK 65000
DIM SHARED İCON(1 TO 675) DIM SHA-
RED YAZS(1 TO 1000)
DIM SHARED MENUİCON1(1 TO 300)

```

```

DIM SHARED MENUİCON2(1 TO 300)
DIM SHARED MENUİCON3(1 TO 300)
DIM SHARED USEFULL(1 TO 1000)
DIM SHARED ALTMENU1TEXT$(3): DIM
SHARED ALTMENU2TEXT$(3)
DIM SHARED ALTMENU3TEXT$(4)
DIM SHARED İNREGS AS REGTYPE,
OUTREGS AS REGTYPE
İNSTALL
CLEAR
İCONMAKE
ON ERROR GOTO D1: OPEN "I", #1,
"PARAM.FLM"
INPUT #1, PARAM$: CLOSE #1 KILL
"PARAM.FLM"
IF PARAM$ = "1" THEN TANİMALT TA-
NİMALT1
YAZSKANT
CLS 2
BULANIKYAZ
SECMENUO
KONTROL1 1
K = 13
D1:
OPEN "O", #1, "GRAPHMAK.FLM"
CLOSE
FİRSTMENU:
RUN "HATAMES.BAS"
UNİNSTALL
REM $STATIC
SUB ALMENU1 (ÇIKİS)
CLOSEMOUSE
KLJ = 8
LINE (100, 31)-(200, 130), 0, BF
LINE (100, 31)-(200, 130), KLJ, B
A = 0
COLOR KLJ
FOR I = 2 TO 3
LOCATE 2 + I, 14
PRINT ALTMENU1TEXT$(I):NEXT
LOCATE 3, 14
PRINT ALTMENU1TEXT$(1)
PAINT (103, 33), 15, KLJ
LINE (101, 40)-(199, 58), 1, B
İNREGS.AX = 1
CALL İNTERRUPT(&H33, İNREGS, OUT-
REGS)
A = 1
ALTMENUBUILD A, KEYBORD$
CLOSE
MOUSE
LINE (40, 0)-(600, 30), 15, BF LINE (100,
31)-(200, 130), 7, BF
IF KEYBORD$ <> CHR$(27) THEN
IF A = 4 OR A = 5 OR A = 6 THEN
IF A = 4 THEN ÇIKİS = 1

```

```

IF A = 5 THEN CİKİS = 2
IF A = 6 THEN CİKİS = 3
GOTO AX1
END IF
X = A / 20 + 4
OPEN "O", #8, "NUMBER.DAT"
WRITE #8, X
CLOSE #8
CHAIN "SECİM1.BAS"
ELSE EXIT SUB
END IF
AX1:
END SUB
SUB ALMENU2 (CİKİS)
CLOSEMOUSE
KLJ = 8
LINE (250, 31)-(350, 100), 0, BF
LINE (250, 31)-(350, 100), KLJ, B
A = 0
COLOR KLJ
FOR I = 2 TO 3
LOCATE 2 + I, 33
PRINT ALTMENU2TEXT$(I)
NEXT
LOCATE 3, 33
PRINT ALTMENU2TEXT$(1)
PAINT (253, 33), 15, KLJ
LINE (251, 40)-(349, 58), 1, B
INREGS.AX = 1
CALL INTERRUPT(&H33, INREGS, OUT-
REGS)
A = 2
ALTMENUBUILD A, KEYBORDS
CLOSEMOUSE
LINE (40, 0)-(600, 30), 15, BF
LINE (250, 31)-(350, 130), 7, BF
IF KEYBORD$ <> CHR$(27) THEN
IF A = 4 OR A = 5 OR A = 6 THEN
IF A = 4 THEN CİKİS = 1
IF A = 5 THEN CİKİS = 2
IF A = 6 THEN CİKİS = 3
GOTO AX2
END IF
X = A / 20 + 4
OPEN "O", #8, "NUMBER.DAT"
WRITE #8, X
CLOSE #8
CHAIN "SECİM2.BAS"
ELSE EXIT SUB
END IF
AX2:
END SUB
SUB ALMENU3 (CİKİS)
CLOSEMOUSE
KLJ = 8
LINE (401, 31)-(500, 130), 0, BF

```

```

LINE (401, 31)-(500, 130), KLJ, B
A = 0
COLOR KLJ
FOR I = 2 TO 4
LOCATE 2 + I, 53
PRINT ALTMENU3TEXT$(I)
NEXT
LOCATE 3, 53
PRINT ALTMENU3TEXT$(1)
PAINT (403, 33), 15, KLJ
LINE (402, 40)-(499, 58), 1, B
INREGS.AX = 1
CALL INTERRUPT(&H33, INREGS, OUT-
REGS)
A = 3
ALTMENUBUILD A, KEYBRD$
KEYBORD$ = KEYBRD$
CLOSEMOUSE
LINE (40, 0)-(600, 30), 15, BF
LINE (400, 31)-(500, 130), 7, BF
IF KEYBORD$ <> CHR$(27) THEN
IF A = 4 OR A = 5 OR A = 6 THEN
IF A = 4 THEN CİKİS = 1
IF A = 5 THEN CİKİS = 2
IF A = 6 THEN CİKİS = 3
GOTO AX3
END IF
SELECT CASE A
CASE 0
RUN "HATAMES.BAS"
CASE 20
SECONDİTEM
CASE 40
UNINSTALL
SCREEN 0
END
CASE 60
SİL
END SELECT
END IF
A3:
END SUB
SUB ALTMENUBUILD (ALTMENUX,
KOD$)
IF ALTMENUX = 1 THEN CX1 = 100 CX2
= 200
IF ALTMENUX = 2 THEN CX1 = 250 CX2
= 350
IF ALTMENUX = 3 THEN CX1 = 400 CX2
= 500
DO
KOD$ = INKEY$
INREGS.AX = 3
CALL INTERRUPT(&H33, INREGS, OUT-
REGS)
IF OUTREGS.BX = 2 THEN

```

```

KOD$ = CHR$(27)
IF OUTREGS.BX = 1 THEN
IF OUTREGS.CX >= CX1 AND OUT-
REGS.CX <= CX2 THEN
IF OUTREGS.DX >= 30 AND OUT-
REGS.DX <= 42 THEN KA = 0: KOD$ =
CHR$(13)
IF OUTREGS.DX >= 45 AND OUT-
REGS.DX <= 57 THEN KA = 20 KOD$ =
CHR$(13)
IF OUTREGS.DX >= 59 AND OUT-
REGS.DX <= 72 THEN KA = 40 KOD$ =
CHR$(13)
IF OUTREGS.CX >= 400 AND OUT-
REGS.CX < 500 AND OUTREGS.DX >= 79
AND OUTREGS.DX <= 92 THEN KA = 60
KOD$ = CHR$(13)
END IF
IF OUTREGS.CX >= 100 AND OUT-
REGS.CX <= 200 AND OUTREGS.DX >=
10 AND OUTREGS.DX <= 28 THEN ALT-
MENUX = 4
KOD$ = CHR$(27)
GOTO JHK
IF OUTREGS.CX >= 250 AND OUT-
REGS.CX <= 350 AND OUTREGS.DX >=
10 AND OUTREGS.DX <= 28 THEN ALT-
MENUX = 5
KOD$ = CHR$(27)
GOTO JHK
IF OUTREGS.CX >= 400 AND OUT-
REGS.CX <= 500 AND OUTREGS.DX >=
10 AND OUTREGS.DX <= 28 THEN ALT-
MENUX = 6
KOD$ = CHR$(27)
GOTO JHK
END IF
LOCATE 10, 1
PRINT OUTREGS.DX, OUTREGS.CX,
OUTREGS.AX, OUTREGS.BX
IF KOD$ = "" THEN GOTO DOA1LOOP
IF KOD$ = CHR$(0) + CHR$(80) THEN
CLOSEMOUSE
B = KA
KA = KA + 20
IF CX1 = 250 THEN IF KA = 40 THEN KA
= 0
IF CX1 = 400 THEN IF KA = 80 THEN KA
= 0
IF CX1 = 100 THEN IF KA = 60 THEN KA
= 0
LINE (CX1 + 2, 40 + KA)-(CX2 - 2, 58 +
KA), 1, B
LINE (CX1 + 2, 40 + B)-(CX2 - 2, 58 + B),
15, B
INREGS.AX = 1

```

```

CALL INTERRUPT(&H33, INREGS, OUT-
REGS)
END IF
IF KOD$ = CHR$(0) + CHR$(72) THEN
CLOSEMOUSE
B = KA
KA = KA - 20
IF CX1 = 250 THEN IF KA = -20 THEN KA
= 20
IF CX1 = 100 THEN IF KA = -20 THEN KA
= 40
IF CX1 = 400 THEN IF KA = -20 THEN KA
= 60

LINE (CX1 + 2, 40 + KA)-(CX2 - 2, 58 +
KA), 1, B
LINE (CX1 + 2, 40 + B)-(CX2 - 2, 58 + B),
15, B
INREGS.AX = 1
CALL INTERRUPT(&H33, INREGS, OUT-
REGS)
END IF
IF CHR$(13) = KOD$ THEN EXIT DO
DOA1LOOP:
LOOP UNTIL KOD$ = CHR$(27)
ALTMENUX = KA
KOD$ = KOD$
JHK:
CLOSEMOUSE
END SUB
SUB BULANIKYAZ
SCREEN 9
VIEW (300, 300)-(340, 320)
WINDOW SCREEN (0, 0)-(640, 480)
VIEW SCREEN (0, 0)-(639, 349)
PUT (600, 0), ICON, PSET
PUT (0, 0), ICON, PSET
PUT (0, 450), ICON, PSET
PUT (600, 450), ICON, PSET
LINE (40, 450)-(600, 480), 15, BF
END SUB
SUB CLOSEMOUSE
INREGS.AX = 2
CALL INTERRUPT(&H33, INREGS, OUT-
REGS)
INREGS.AX = &H10
INREGS.CX = 0
INREGS.DI = 349
INREGS.DX = 0
INREGS.SI = 639
CALL INTERRUPT(&H33, INREGS, OUT-
REGS)
END SUB
SUB FIRSTMENU
TANIMALT
TANIMALT1

```

```

YAZSKANT
SECONDITEM
END SUB
SUB FRAMEICON (X1, X2, Y1, Y2,
BACKGROUND)
LINE (X1, Y1)-(X2, Y2), BACKGROUND,
BF
END SUB
SUB ICONMAKE
DEF SEG = VARSEG(ICON(1))
BLOAD "ICON.FLM", VARPTR(ICON(1))
DEF SEG
END SUB
SUB INSTALL
INREGS.AX = &H1: CALL INTER-
RUPT(&H33, INREGS, OUTREGS)
END SUB
SUB KONTR0L1 (KJ)
SECMENUO
MENUINSTALL 400, 250
MENUINSTALL 250, 100
VIEW PRINT 3 TO 22
ICONMAKE
KEYA = 100
KEYB = KEYA
INREGS.AX = 1
CALL INTERRUPT(&H33, INREGS, OUT-
REGS)
DO
AX:
INREGS.AX = 1
CALL INTERRUPT(&H33, INREGS, OUT-
REGS)
KEYBORD$ = INKEY$
INREGS.AX = 3
CALL INTERRUPT(&H33, INREGS, OUT-
REGS)
IF OUTREGS.BX = 1 THEN
IF OUTREGS.DX >= 7 AND
OUTREGS.DX <= 20 THEN
IF OUTREGS.CX >= 99 AND OUT-
REGS.CX <= 178 THEN CLOSEMOUSE
MENUINSTALL KEYB, 100
INREGS.AX = &H1
CALL INTERRUPT(&H33, INREGS, OUT-
REGS)
KEYB = 100
KEYBORD$ = CHR$(13)
IF OUTREGS.CX >= 249 AND OUT-
REGS.CX <= 328 THEN CLOSEMOUSE
MENUINSTALL KEYB, 250
INREGS.AX = &H1
CALL INTERRUPT(&H33, INREGS, OUT-
REGS)
KEYB = 250
KEYBORD$ = CHR$(13)

```

```

IF OUTREGS.CX >= 399 AND OUT-
REGS.CX <= 477 THEN CLOSEMOUSE
MENUINSTALL KEYB, 400
INREGS.AX = &H1
CALL INTERRUPT(&H33, INREGS, OUT-
REGS)
KEYB = 400
KEYBORD$ = CHR$(13)
END IF
END IF

IF KEYBORD$ = "" THEN GOTO DONH
IF KEYBORD$ = CHR$(0) + CHR$(75)
THEN
KEYA = KEYB
KEYB = KEYB - 150
IF KEYB <= 0 THEN KEYB = 400
MENUINSTALL KEYA, KEYB
END IF
IF KEYBORD$ = CHR$(0) + CHR$(77)
THEN
KEYA = KEYB
KEYB = KEYB + 150
IF KEYB = 550 THEN KEYB = 100
MENUINSTALL KEYA, KEYB
END IF
IF KEYBORD$ = CHR$(13) THEN
SELECT CASE KEYB
CASE 100
ALMENU1 GF
IF GF = 4 OR GF = 5 OR GF = 6 THEN
GOTO AX
MENUINSTALL 400, 250
MENUINSTALL 250, 100
KEYB = 100
CASE 250
ALMENU2 GF
IF GF = 4 OR GF = 5 OR GF = 6 THEN
GOTO AX
KEYB = 250
KEYA = 100
MENUINSTALL KEYA, KEYB
MENUINSTALL 400, KEYB
CASE 400
ALMENU3 GF
IF GF = 4 OR GF = 5 OR GF = 6 THEN
GOTO AX
KEYB = 400
KEYA = 100
MENUINSTALL KEYA, KEYB
MENUINSTALL 250, KEYB
END SELECT
END IF
INREGS.AX = 1
CALL INTERRUPT(&H33, INREGS, OUT-
REGS)

```



```

DONH:
LOOP UNTIL KEYBOARD$ = CHR$(27)
CLOSEMOUSE
SECONDIEM
END SUB
SUB KONTROLANA
F1$ = CHR$(0) + CHR$(59)
F2$ = CHR$(0) + CHR$(60)
F3$ = CHR$(0) + CHR$(61)
F4$ = CHR$(0) + CHR$(62)
INREGS.AX = 1
CALL INTERRUPT(&H33, INREGS, OUT-
REGS)
A1:
TEXT$ = INKEY$
INREGS.AX = 3
CALL INTERRUPT(&H33, INREGS, OUT-
REGS)
IF OUTREGS.BX = 1 THEN
IF OUTREGS.DX >= 329 AND OUT-
REGS.DX <= 347 THEN
IF OUTREGS.CX >= 46 AND OUT-
REGS.CX <= 129 THEN TEXT$ = F1$
IF OUTREGS.CX >= 176 AND OUT-
REGS.CX <= 289 THEN TEXT$ = F2$
IF OUTREGS.CX >= 348 AND OUT-
REGS.CX <= 474 THEN TEXT$ = F3$
IF OUTREGS.CX >= 508 AND OUT-
REGS.CX <= 585 THEN TEXT$ = F4$
END IF
END IF

IF TEXT$ = "" THEN GOTO A1
SELECT CASE TEXT$
CASE F1$
BEEP
BEEP
CASE F2$
CLOSEMOUSE
RUN "HATAMES"
CASE F3$
MESAJ
CLS 2
PAINT (200, 200), 7
CLOSEMOUSE
BULANIKYAZ
SECMENUO
KONTROL1 (1)
CASE F4$
SCREEN 0, 0
UNINSTALL
CLOSEMOUSE
END
CASE ELSE
BEEP
END SELECT

```

```

GOTO A1
END SUB
SUB MENUINSTALL (K1, K2)
LINE (K1, 10)-(K1 + 78, 28), 15, BF
SELECT CASE K1
CASE 100
PUT (K1, 10), MENUICON1, XOR
CASE 250
LINE (K1, 10)-(K1 + 100, 28), 15, BF
PUT (K1, 10), MENUICON2, XOR
CASE 400
PUT (K1, 10), MENUICON3, XOR
END SELECT
SELECT CASE K2
CASE 100
PUT (K2, 10), MENUICON1, PSET
CASE 250
PUT (K2, 10), MENUICON2, PSET
CASE 400
PUT (K2, 10), MENUICON3, PSET
END SELECT
END SUB
SUB MESAJ
CLOSEMOUSE
COLOR 15, 9
LINE (0, 0)-(690, 480), 7, BF
LINE (40, 40)-(600, 300), 14, BF
LINE (56, 120)-(576, 350), 15, BF
LINE (56, 350)-(576, 380), 10, BF
LOCATE 8, 14
PRINT "BU BOLUMDE YENİ BİR SİSTEM
TANIMLAMASI YAPILIR. "
LOCATE 9, 14
PRINT "EĞER YENİ BİR SİSTEM KONT-
ROLU YAPACAKSANIZ EVET "
LOCATE 10, 14
PRINT "SECENEGİNİ ( E TUSUY-
LA), İSTEMİYORSANIZ HAYİR SE- "
LOCATE 11, 14
PRINT " CENEGİNİ ( H TUSUYLA) SE-
CİNİZ."
LOCATE 12, 14
PRINT " !!! "
LOCATE 13, 14
PRINT "EVET SECENEGİNİ SECTİĞİNİZ
TAKTİRDE BİR ÖNCEKİ SİS-"
LOCATE 14, 14
PRINT "TEME AİT TUM VERİLER SİLİ-
NECEKTİR."
LOCATE 19, 8
COLOR 12, 4
PRINT " <EVET>
<HAYIR> "
COLOR 15, 9
INREGS.AX = 1

```

```

CALL INTERRUPT(&H33, INREGS, OUT-
REGS)
TRE:
DE$ = INKEY$
INREGS.AX = 3
CALL INTERRUPT(&H33, INREGS, OUT-
REGS)
IF OUTREGS.BX = 1 THEN
IF OUTREGS.DX >= 252 AND OUT-
REGS.DX <= 266 THEN
IF OUTREGS.CX >= 66 AND OUT-
REGS.CX <= 146 THEN DE$ = "E"
IF OUTREGS.CX >= 476 AND OUT-
REGS.CX <= 537 THEN DE$ = "H"
END IF
END IF
IF INREGS.AX THEN
IF DE$ = "E" OR DE$ = "H" THEN
COLOR 7, 0
EXIT SUB
ELSEIF DE$ = "H" OR DE$ = "H" THEN
COLOR 7, 0
CLOSEMOUSE
FIRSTMENU
ELSE
GOTO TRE
END IF
END SUB
SUB QWAIT
1:
WAITQ$ = INKEY$
IF NOT WAITQ$ = "" THEN EXIT SUB
GOTO 1
END SUB
PUT (100, 10), MENUICON1, PSET
PUT (250, 10), MENUICON2, XOR ,
PUT (400, 10), MENUICON3, XOR
BULANIKYAZ
KONTROL1 1
END SUB
SUB SECMENUO
COLOR 14
VIEW PRINT 1 TO 23
LOCATE 1, 10
PRINT "< TANIM1 >"
LOCATE 1, 30
PRINT "< TANIM2 >"
LOCATE 1, 50
PRINT "< YURUTME >"
GET (393, 0)-(471, 18), MENUICON3
GET (232, 0)-(310, 18), MENUICON2
GET (72, 0)-(150, 18), MENUICON1
LINE (40, 0)-(600, 30), 15, BF
VIEW PRINT 3 TO 23
COLOR 7
LINE (0, 30)-(639, 450), 7, BF

```

```

END SUB
SUB SECONDIEM
SCREEN 9
WINDOW SCREEN (0, 0)-(640, 480)
YAZI (3)
BULANIKYAZ
PUT (44, 460), YAZS, PRESET
INREGS.AX = 1
CALL INTERRUPT(&H33, INREGS, OUT-
REGS)
KONTROLANA
END SUB
SUB SIL
LINE (150, 150)-(400, 300), 14, BF
LINE (150 + 5, 150 + 5)-(400 - 5, 300 - 5), 7,
BF
COLOR , 7
LOCATE 4, 20: PRINT "SILME BOLUMU"
OPEN "I", #1, "CALISMAR.FLM"
OPEN "O", #2, "GECIS.FLG"
DO
SAYAC = SAYAC + 1
INPUT #1, A$
IF SAYAC = 6 THEN SAYAC = 1 KAYAC =
KAYAC + 10
LOCATE 9 + SAYAC, 22 + KAYAC PRINT
A$
LOOP UNTIL EOF(1)
LOCATE 17, 22
INPUT "SILINECEK ARAC ADI...>", A$
A$ = UCASE$(A$)
SEEK #1, 1
DO
INPUT #1, DOSYA$
IF DOSYA$ = A$ THEN KILL DOSYA$ +
" ."
GOTO D
WRITE #2, DOSYA$
D:
LOOP UNTIL EOF(1)
CLOSE
LINE (150, 150)-(400, 300), 7, BF
CLS 2
COLOR , 0
LINE (0, 35)-(639, 450), 7, BF
KILL "CALISMAR.FLM"
NAME "GECIS.FLG" AS
"CALISMAR.FLM"
END SUB
SUB SILME
REM *****UYELIK FONKSIYON-
LARININ SILINMESI*****
OPEN "O", #1, "UFKAYIT.DAT"
REDIM UFG1$(10)
REDIM UFG2$(10)
REDIM UFG3$(10)

```

```

FOR I = 1 TO 10
  UFG1$(I) = ""
  UFG2$(I) = ""
  UFG3$(I) = ""
  WRITE #1, UFG1$(I), UFG2$(I), UFG3$(I)
NEXT I
UFC$ = ""
WRITE #1, UFC$
CLOSE #1
ERASE UFG1$
ERASE UFG2$
ERASE UFG3$
OPEN "O", #4, "DEGER.DAT"
OPEN "O", #5, "DEGER1.DAT"
REDIM K11(10)
REDIM K12(10)
REDIM K13(10)
FOR I = 1 TO 10
  K11(I) = 2
  K12(I) = 2
  K13(I) = 2
  WRITE #5, K11(I), K12(I), K13(I)
NEXT I
CLOSE #5
REDIM C11$(10, 2)
REDIM C12$(10, 2)
REDIM C13$(10, 2)
FOR K = 1 TO 10
  FOR I = 1 TO K11(K)
    C11$(K, I) = ""
    WRITE #4, C11$(K, I)
  NEXT I
  FOR I = 1 TO K12(K)
    C12$(K, I) = ""
    WRITE #4, C12$(K, I)
  NEXT I
  FOR I = 1 TO K13(K)
    C13$(K, I) = ""
    WRITE #4, C13$(K, I)
  NEXT I
NEXT K
CLOSE #4
ERASE K11: ERASE K12
ERASE K13
ERASE C11$: ERASE C12$ ERASE C13$
REM UYELIK KUMELERININ SILINMESI
OPEN "O", #2, "UKKAYIT.DAT"
OPEN "O", #3, "SAYI.DAT"
REDIM GUK(10)
REDIM GUK$(10, 7)
REDIM CUK$(7)
FOR I = 1 TO 10
  GUK(I) = 1
  WRITE #3, GUK(I)
NEXT I
CUK = 1

```

```

WRITE #3, CUK
CLOSE #3
FOR I = 1 TO 10
  FOR K = 1 TO GUK(I)
    GUK$(I, K) = ""
    WRITE #2, GUK$(I, K)
  NEXT K
NEXT I
FOR I = 1 TO CUK
  CUK$(I) = ""
  WRITE #2, CUK$(I)
NEXT I
CLOSE #2
ERASE GUK
ERASE GUK$
ERASE CUK$
REM *****SABITLERIN SILIN-
MESI*****
OPEN "O", #6, "SABITGC.DAT"
REDIM SA(10, 2, 2)
REDIM SA1(10, 2, 2)
REDIM SA2(10, 2, 2)
REDIM SAC(1, 2)
FOR I = 1 TO 10
  FOR K = 1 TO 2
    SA(I, 1, K) = 0
    WRITE #6, SA(I, 1, K)
  NEXT K
  FOR K = 1 TO 2
    SA1(I, 1, K) = 0
    WRITE #6, SA1(I, 1, K)
  NEXT K
  FOR K = 1 TO 2
    SA2(I, 1, K) = 0
    WRITE #6, SA2(I, 1, K)
  NEXT K
NEXT I
SAC(1, 1) = 0
WRITE #6, SAC(1, 1)
CLOSE #6
ERASE SA
ERASE SA1
ERASE SA2
ERASE SAC
REM ARALIK DEGERLERININ SILIN-
MESI
OPEN "O", #7, "SINIR.DAT"
REDIM ARA1(10, 2, 2)
REDIM ARA2(10, 2, 2)
REDIM ARA3(10, 2, 2)
FOR A = 1 TO 10
  FOR K = 1 TO 2
    ARA1(A, 1, K) = 0
    WRITE #7, ARA1(A, 1, K)
  NEXT K
  FOR K = 1 TO 2

```

```

ARA2(A, 1, K) = 0
WRITE #7, ARA2(A, 1, K)
NEXT K
FOR K = 1 TO 2
ARA3(A, 1, K) = 0
WRITE #7, ARA3(A, 1, K)
NEXT K
NEXT A
CLOSE #7
ERASE ARA1
ERASE ARA2
ERASE ARA3
REM *****KURALLARIN SILINME-
SI*****
REDIM KURAL$(9, 10, 1)
OPEN "O", #12, "KURALL.DAT"
FOR A = 1 TO 9
FOR B = 1 TO 10
KURAL$(A, B, 1) = ""
WRITE #12, KURAL$(A, B, 1)
NEXT B
NEXT A
CLOSE #12
ERASE KURALS
END SUB
SUB TANIMALT
ALTMENU1TEXT$(1) = "FONKSİYON"
ALTMENU1TEXT$(2) = "KUMELER"
ALTMENU1TEXT$(3) = "SABİTLER"
ALTMENU3TEXT$(2) = "ANAMENU"
ALTMENU3TEXT$(1) = "CALIŞTIR"
ALTMENU3TEXT$(3) = "ÇIKIŞ"
ALTMENU3TEXT$(4) = "SİLME"
END SUB
SUB TANIMALT1
ALTMENU2TEXT$(1) = "ARALIK"
ALTMENU2TEXT$(2) = "KURAL.T"
END SUB
SUB TUS
DO
IS = INKEY$: LOOP UNTIL IS <> ""
END SUB
SUB UNINSTALL
SHELL "COPY D:\FLM\*. * C:\FLM\*. *
>>NUL"
SHELL "D:"
SHELL "CD \"
SHELL "CD FLM"
SCREEN 0
PRINT "DEVAM ETMEK İÇİN 'Y' VE 'EN-
TER'E BASIN "
SHELL "DEL D:\FLM\*. *>>NUL"
SHELL "CD \"
SHELL "RD FLM"
SHELL "C:"
END SUB

```

```

SUB YARDIM
LOCATE 4, 35
PRINT "YARDIM MENUSU"
END SUB
SUB YAZI (C)
PAINT (1, 1), 7
VIEW PRINT 1 TO 25
CLS 2
A = 10
B = 101
DUSEY = 20
SAHA = 60
XY = -25
YX = 50
SKL = 640
SKK = 465
CIRCLE STEP(341 + XY, 160 + YX), 300, 1,
, , -9.32
PAINT (330 + XY, 160 + YX), 1
LINE (SKL, SKK)-(160 - 20, 300 + DUSEY),
C
LINE (SKL, SKK)-(230 - 20, 300 + DUSEY),
C
LINE (SKL, SKK)-(195 - 20, 340 + DUSEY),
C
LINE (SKL, SKK)-(160 - 20, 400 + DUSEY),
C
LINE (SKL, SKK)-(230 - 20, 400 + DUSEY),
C
LINE (SKL, SKK)-(240 - 20, 400 + DUSEY),
C
LINE (SKL, SKK)-(310 - 20, 400 + DUSEY),
C
LINE (SKL, SKK)-(255 - 20, 360 + DUSEY),
C
LINE (SKL, SKK)-(295 - 20, 360 + DUSEY),
C
LINE (SKL, SKK)-(275 - 20, 300 + DUSEY),
C
LINE (SKL, SKK)-(320 - 20, 400 + DUSEY),
C
LINE (SKL, SKK)-(320 - 20, 300 + DUSEY),
C
LINE (SKL, SKK)-(390 - 20, 400 + DUSEY),
C
LINE (SKL, SKK)-(390 - 20, 300 + DUSEY),
C
LINE (SKL, SKK)-(400 - 20, 300 + DUSEY),
C
LINE (SKL, SKK)-(435 - 20, 300 + DUSEY),
C
LINE (SKL, SKK)-(470 - 20, 300 + DUSEY),
C
LINE (SKL, SKK)-(435 - 20, 400 + DUSEY),
C

```

```

LINE (SKL, SKK)-(480 - 20, 300 + DUSEY),
C
LINE (SKL, SKK)-(480 - 20, 400 + DUSEY),
C
LINE (SKL, SKK)-(490 - 20, 300 + DUSEY),
C
LINE (SKL, SKK)-(560 - 20, 300 + DUSEY),
C
LINE (SKL, SKK)-(490 - 20, 400 + DUSEY),
C
LINE (SKL, SKK)-(560 - 20, 400 + DUSEY),
C
LINE (SKL, SKK)-(490 - 20, 370 + DUSEY),
C
FOR I = -25 TO -20
LINE (160 + I, 300 + DUSEY)-(160 + I, 400
+ DUSEY), B
LINE (230 + I, 300 + DUSEY)-(230 + I, 400
+ DUSEY), B
LINE (160 + I, 300 + DUSEY)-(195 + I, 340
+ DUSEY), B
LINE (195 + I, 340 + DUSEY)-(230 + I, 300
+ DUSEY), B
LINE (240 + I, 400 + DUSEY)-(275 + I, 300
+ DUSEY), B
LINE (275 + I, 300 + DUSEY)-(310 + I, 400
+ DUSEY), B
LINE (255 + I, 360 + DUSEY)-(295 + I, 360
+ DUSEY), B
LINE (320 + I, 300 + DUSEY)-(320 + I, 400
+ DUSEY), B
LINE (390 + I, 300 + DUSEY)-(390 + I, 400
+ DUSEY), B
LINE (320 + I, 300 + DUSEY)-(390 + I, 400
+ DUSEY), B
LINE (400 + I, 300 + DUSEY)-(470 + I, 300
+ DUSEY), B
LINE (435 + I, 300 + DUSEY)-(435 + I, 400
+ DUSEY), B
LINE (480 + I, 300 + DUSEY)-(480 + I, 400
+ DUSEY), B
LINE (490 + I, 300 + DUSEY)-(490 + I, 400
+ DUSEY), B
LINE (560 + I, 300 + DUSEY)-(490 + I, 370
+ DUSEY), B
LINE (510 + I, 350 + DUSEY)-(560 + I, 400
+ DUSEY), B
NEXT
FOR I = -35 TO -30
FOR AHA = -5 TO 0
SAHA = AHA * AHA
CIRCLE (100 + I, 100 + SAHA), 50, A, 4.64,
1.6
CIRCLE (100 + I, 200 + SAHA), 50, A, 4.64,
1.5

```

```

LINE (98 + I, 50 + SAHA)-(98 + I, 250 +
SAHA), A
CIRCLE (210 + I, 200 + SAHA), 50, A, 3.14,
0
LINE (160 + I, 50 + SAHA)-(160 + I, 200 +
SAHA), A
LINE (260 + I, 50 + SAHA)-(260 + I, 200 +
SAHA), A
LINE (270 + I, 50 + SAHA)-(270 + I, 250 +
SAHA), A
LINE (270 + I, 250 + SAHA)-(340 + I, 250 +
SAHA), A
LINE (385 + I, 50 + SAHA)-(350 + I, 250 +
SAHA), A
LINE (385 + I, 50 + SAHA)-(420 + I, 250 +
SAHA), A
LINE (360 + I, 200 + SAHA)-(410 + I, 200 +
SAHA), A
LINE (430 + I, 50 + SAHA)-(430 + I, 250 +
SAHA), A
LINE (500 + I, 50 + SAHA)-(500 + I, 250 +
SAHA), A
LINE (430 + I, 50 + SAHA)-(500 + I, 250 +
SAHA), A
LINE (510 + I, 50 + SAHA)-(510 + I, 250 +
SAHA), A
LINE (520 + I, 50 + SAHA)-(520 + I, 250 +
SAHA), A
LINE (520 + I, 200 + SAHA)-(595 + I, 50 +
SAHA), A
LINE (540 + I, 160 + SAHA)-(595 + I, 250 +
SAHA), A
NEXT
NEXT
END SUB
SUB YAZSKANT
SCREEN 9, 0
WIDTH 80, 25
CLS
VIEW (300, 300)-(340, 320)
WINDOW SCREEN (0, 0)-(640, 480)
VIEW SCREEN (0, 0)-(639, 349)
LOCATE 1, 1: PRINT "<F1-YARDIM>"
<F2-ESKİ SİSTEM> <F3-YENİ SİS-
TEM> <F4-ÇIKIŞ> "GET (0,
0)-(550, 18), YAZS
END SUB

```

*****SEÇİMİ*****

" Bu bölümde denetlenecek sisteme ait üyelik fonksiyonları, kümeler ve sabitlerin tanımlanması yapılmaktadır."

DECLARE SUB OKU1 () DECLARE SUB YMENU ()

```

DECLARE SUB TANIM () DECLARE SUB
TUS ()
DECLARE SUB STDMENU () DECLARE
SUB SEC1 ()
DECLARE SUB QINPUT (X!, Y!, STRNG$,
NUMERIC!, DIMENSION!, KODE$)
DECLARE SUB OKU1 ()
'QBX.BI
COMMON SHARED KATAR$, ESC$,
BACKSPACES$, INSERT$,
UP$, DOWN$, LLEFT$, RRIGHT$, SPCE$,
ENTERS$, F1$,
F2$, F3$, F4$, F5$, F6$, F7$, F8$, F9$, F10$
COMMON SHARED D, K, X, Y, CUK, S, L
DIM SHARED C11$(10, 1 TO 20): DIM
SHARED C12$(10, 1 TO 20)
DIM SHARED C13$(10, 1 TO 20) DIM
SHARED GUK(10)
DIM SHARED K11(10): DIM SHARED
K12(10)
DIM SHARED K13(10): DIM SHARED
UFG1$(10)
DIM SHARED UFG2$(10): DIM SHARED
UFG3$(10)
DIM SHARED GUK$(10, 1 TO 7)
DIM SHARED SAC(1, 1 TO 7)
DIM SHARED SA1(10, 1 TO 7, 2)
DIM SHARED SA2(10, 1 TO 7, 2)
DIM SHARED CUK$(1 TO 7)
DIM SHARED INREGS AS REGTYPE,
OUTREGS AS REGTYPE
INREGS.AX = &H1
CALL INTERRUPT(&H33, INREGS, OUT-
REGS)
INREGS.AX = &H4
INREGS.CX = 632
INREGS.DX = 192
CALL INTERRUPT(&H33, INREGS, OUT-
REGS)
INREGS.AX = &H3
SEC1
OPEN "O", #1, "PARAM.FLM"
WRITE #1, "1"
CLOSE
RUN "36A.BAS"

SUB OKU1
CK = 1
CLS 2
COLOR 0, 7
OPEN "I", #1, "UFKAYIT.DAT"
OPEN "I", #3, "SAYI.DAT"
OPEN "I", #4, "DEGER.DAT"
OPEN "I", #5, "DEGER1.DAT"
OPEN "O", #6, "SABITGC.DAT"

```

```

M = 10
FOR I = 1 TO 10
INPUT #1, UFG1$(I), UFG2$(I), UFG3$(I)
INPUT #5, K11(I), K12(I), K13(I)
INPUT #3, GUK(I)
FOR K = 1 TO K11(I)
INPUT #4, C11$(I, K)
NEXT K
FOR K = 1 TO K12(I)
INPUT #4, C12$(I, K)
NEXT K
FOR K = 1 TO K13(I)
INPUT #4, C13$(I, K)
NEXT K
NEXT I
INPUT #3, CUK
INPUT #1, UFC$
CLOSE #1, #3, #4, #5
CLS 2
CALL STDMENU
FOR K = 1 TO 10
S = 3
LOCATE S, 8: COLOR 14
PRINT "UYELIK FONKSIYONLARI"
LOCATE S + 1, 8
PRINT "-----"
LOCATE S + 2, 3: COLOR 14 PRINT K;
"GIRIS ", "1. ARALIKTA U.F..."; UFG1$(K)
LOCATE S + 4, 3: COLOR 14: PRINT K;
"GIRIS ", "2. ARALIKTA U.F..."; UFG2$(K)
LOCATE S + 6, 3: COLOR 14 PRINT K;
"GIRIS ", "3. ARALIKTA U.F..."; UFG3$(K)
LOCATE S, 61: COLOR 14 PRINT
"G.SABITLERI "
LOCATE S + 1, 61: COLOR 14 PRINT "1.A-
2.A-3.A "
LOCATE S + 2, 61: COLOR 14 PRINT "-----"
S = 5
FOR I = 1 TO K11(K)
S = S + 1
LOCATE S, 62: COLOR 14, 7 PRINT
C11$(K, I)
NEXT I
S = 5
FOR I = 1 TO K12(K)
S = S + 1
LOCATE S, 66: COLOR 14, 7 PRINT
C12$(K, I)
NEXT I
S = 5
FOR I = 1 TO K13(K)
S = S + 1
LOCATE S, 70: COLOR 14, 7 PRINT
C13$(K, I)
NEXT I

```

EK-1

```

LOCATE 3, 7: PRINT "GIRIS ICIN TA-
NIMLANMIS UYELIK KUMELERINE AIT
SABITLERIN TANIMLANMASI"
S = 15
FOR I = 1 TO GUK(K)
FOR KL = 1 TO K11(K)
L = 5
LOCATE 14, 5: COLOR 14, 7: PRINT "
1.ARALIK ICIN..: "
LOCATE S, L: COLOR 14, 7: PRINT " "; K;
" "; "GIRIS"; I, " "; "UYELIK KUMESINE
AIT"; " "; " "; "SABITI..:"
LOCATE S, 40: COLOR 14, 7: PRINT
C11$(K, KL)
QINPUT 50, S, A$, SA(K, I, KL), 5, KOD$
LOCATE S, 50: PRINT " "
NEXT KL
FOR KL = 1 TO K12(K)
LOCATE 14, 5: COLOR 14, 7: PRINT "
2.ARALIK ICIN..: "
L = 5
LOCATE S, L: COLOR 14, 7: PRINT " "; K;
" "; "GIRIS"; I, " "; "UYELIK KUMESINE
AIT"; " "; " "; "SABITI..:"
LOCATE S, 40: COLOR 14, 7: PRINT
C12$(K, KL)
QINPUT 50, S, A$, SA1(K, I, KL), 5, KOD$
LOCATE S, 50: PRINT " "
NEXT KL
FOR KL = 1 TO K13(K)
LOCATE 14, 5: COLOR 14, 7: PRINT "
3.ARALIK ICIN..: "
LOCATE S, L: COLOR 14, 7
PRINT " "; K; " "; "GIRIS"; I, " "; "UYELIK
KUMESINE AIT"; " "; " "; "SABITI..:"
LOCATE S, 40: COLOR 14, 7: PRINT
C13$(K, KL)
QINPUT 50, S, A$, SA2(K, I, KL), 5, KOD$
LOCATE S, 50: PRINT " "
NEXT KL
NEXT I
Q = 5
FOR I = 1 TO K11(K)
Q = Q + 1
LOCATE Q, 62: COLOR 14, 7: PRINT " "
NEXT I
Q = 5
FOR I = 1 TO K12(K)
Q = Q + 1
LOCATE Q, 66: COLOR 14, 7: PRINT " "
NEXT I
Q = 5
FOR I = 1 TO K13(K)
Q = Q + 1
LOCATE Q, 70: COLOR 14, 7: PRINT " "
NEXT I

```

```

S = 3
LOCATE S + 2, 3: COLOR 14, 7: PRINT "
"
LOCATE S + 4, 3: COLOR 14, 7: PRINT "
"
LOCATE S + 6, 3: COLOR 14, 7: PRINT "
"
NEXT K
L = 5
CK = 1
LOCATE 3, 7: COLOR 14
PRINT "CIKIS ICIN TANIMLANMIS UYE-
LIK KUMELERINE SABITLERIN TANIM-
LANMASI "
S = 5
LOCATE S + 1, 8: COLOR 14, 7: PRINT
"CIKIS ICIN U.F.."; UFC$
LOCATE 3, 61: COLOR 14, 7: PRINT
"□.SABITLERI "
LOCATE 4, 61: COLOR 14, 7: PRINT "
"
LOCATE S + 1, 67: COLOR 14, 7: PRINT
UFC$
LOCATE 14, 5: COLOR 14, 7: PRINT "
"
LOCATE 15, 5: COLOR 14, 7: PRINT "
"
S = 15
FOR I = 1 TO CUK
LOCATE S, L: COLOR 14
PRINT " "; "CIKIS"; I, " "; "UYELIK KU-
MESINE AIT"; " "; " "; "SABITI..:"
LOCATE S, 37: COLOR 14, 7: PRINT UFC$
QINPUT 48, S, A$, SAC(CK, I), 5, KOD$
LOCATE S, 48: PRINT " "
NEXT I
FOR K = 1 TO 10
FOR I = 1 TO GUK(K)
FOR KL = 1 TO K11(K)
WRITE #6, SA(K, I, KL)
NEXT KL
FOR KL = 1 TO K12(K)
WRITE #6, SA1(K, I, KL)
NEXT KL
FOR KL = 1 TO K13(K)
WRITE #6, SA2(K, I, KL)
NEXT KL
NEXT I
NEXT K
FOR I = 1 TO CUK
WRITE #6, SAC(CK, I)
NEXT I
CLOSE #6
KLM:
LOCATE S, 5: COLOR 14, 7: PRINT "
"

```



```

LOCATE S, 3: COLOR 14, 7 PRINT "
KAYIT TAMAMLANMISTIR. MENU ICIN
HER HANGI BIR TUSA BASINIZ.
"
CALL TUS
CLS 2
COLOR 0, 7
END SUB
SUB QINPUT (X, Y, STRNG$, NUMERIC,
DIMENSION, KODE$)
KPKL2 = Y
KPKL4 = X
COLOR 15
TANIM
STRNG$ = ""
NUMERIC = 0
KODE$ = ""
DO
ASS$ = INKEY$
K = K + 1
IF K = 5 THEN LOCATE KPKL2, KPKL4:
PRINT STRNG$; "Ü"
IF K = 10 THEN LOCATE KPKL2, KPKL4:
PRINT STRNG$; " ": K = 0
IF ASS$ = "" THEN GOTO DON
SELECT CASE ASS$
CASE ESC$
KODE$ = ESC$
GOTO DONSON
CASE BACKSPACES$
ASİ = LEN(STRNG$)
IF ASİ = 0 THEN BEEP
GOTO DON
LOCATE KPKL2, KPKL4
PRINT SPACES$(ASİ + 1)
STRNG$ = LEFT$(STRNG$, ASİ - 1)
NUMERIC = VAL(STRNG$)
LOCATE KPKL2, KPKL4
PRINT STRNG$ + "Ü"
LOCATE KPKL2, KPKL4: PRINT STRNG$;
" "
CASE UP$
KODE$ = ASS$
GOTO DONSON
CASE DOWN$
KODE$ = ASS$
GOTO DONSON
CASE LLEFT$
KODE$ = ASS$
GOTO DONSON
CASE RRIGHT$
KODE$ = ASS$
GOTO DONSON
CASE SPCE$
CASE ENTER$
KODE$ = ENTER$

```

```

GOTO DONSON
CASE F1$
KODE$ = ASS$
GOTO DONSON
CASE F2$
KODE$ = ASS$
GOTO DONSON
CASE F3$
KODE$ = ASS$
GOTO DONSON
CASE F4$
KODE$ = ASS$
GOTO DONSON
CASE F5$
KODE$ = ASS$
GOTO DONSON
CASE F6$
KODE$ = ASS$
GOTO DONSON
CASE F7$
KODE$ = ASS$
GOTO DONSON
CASE F8$
KODE$ = ASS$
GOTO DONSON
CASE F9$
KODE$ = ASS$
GOTO DONSON
CASE F10$
KODE$ = ASS$
GOTO DONSON
CASE ELSE
IF LEN(STRNG$) = DIMENSION AND ASS$
< "" THEN BEEP
GOTO DON
IF ASS$ < CHR$(32) THEN GOTO DON
STRNG$ = STRNG$ + ASS$
LOCATE KPKL2, KPKL4
PRINT STRNG$; "Ü"
LOCATE KPKL2, KPKL4
PRINT STRNG$; " "
NUMERIC = VAL(STRNG$)
END SELECT
DON:
LOOP UNTIL KODE$ = ESC$
DONSON:
LOCATE KPKL2, KPKL4
PRINT STRNG$; " "
END SUB
SUB SEC1
OPEN "I", #8, "NUMBER.DAT"
INPUT #8, X
CLOSE #8
Z = X - 3
SELECT CASE Z
CASE 1

```



```

CLS 2
OPEN "O", #1, "UFKAYIT.DAT"
OPEN "O", #4, "DEGER.DAT"
OPEN "O", #5, "DEGER1.DAT"
REDIM C1$(10, 20)
REDIM C2$(10, 20)
REDIM C3$(10, 20)
FOR K = 1 TO 10
CLS 2
LOCATE 5, 3: COLOR 14, 7 PRINT "****";
K; ". GIRIS ICIN YEN ŞYEL K
FONKS YONLARI****"
LOCATE 10, 3
PRINT "1. ARA LIKTA TANIMLANACAK
UYELIK FONKS YONU...": QINPUT 49,
10, UFG1$(K), A, 30, KOD$
LOCATE 11, 3
PRINT "2. ARA LIKTA TANIMLANACAK
UYELIK FONKS YONU...": QINPUT 49,
11, UFG2$(K), A, 30, KOD$
LOCATE 12, 3: PRINT "3. ARA LIKTA TA-
NIMLANACAK UYELIK FONKS YONU..."
QINPUT 49, 12, UFG3$(K), A, 30, KOD$
NEXT K
LOCATE 14, 10: PRINT "IKI N
ŞYEL K FONKS YONU...": QINPUT 43,
14, UFC$, A, 1, KOD$
FOR K = 1 TO 10
DO
CLS 2
LOCATE 3, 20: PRINT K; ".G-"; "1. ARA-
LIKTA TANIMLANAN UYELIK
FONKS YONU"
LOCATE 5, 20: PRINT K; ".G-"; "2. ARA-
LIKTA TANIMLANAN UYELIK
FONKS YONU"
LOCATE 7, 20: PRINT K; ".G-"; "3. ARA-
LIKTA TANIMLANAN UYELIK
FONKS YONU"
LOCATE 10, 30: PRINT "IKI N
ŞYEL K FONKS YONU"
LOCATE 4, 40: PRINT UFG1$(K)
LOCATE 6, 40: PRINT UFG2$(K)
LOCATE 8, 40: PRINT UFG3$(K)
UZ = LEN(UFG1$(K))
A1 = 0
FOR I = 40 TO UZ + 40
B1$ = CHR$(SCREEN(4, I))
IF B1$ < "(" AND B1$ < ")" AND B1$ <
"X" AND B1$ < "X" AND B1$ < "-" AND
B1$ < "+" AND B1$ < "*" AND B1$ <
"/" AND B1$ < "^" AND B1$ < "1" AND
B1$ < "2" AND B1$ < "3" AND B1$ <
"4" AND B1$ < "5" AND B1$ < "6" AND
B1$ < "7" AND B1$ < "8" AND B1$ <
"9" THEN

```

```

A1 = A1 + 1
C1$(K, A1) = B1$
ELSE
GOTO M3
END IF
M3:
NEXT I
Q1 = 30: L1 = A1
I1 = 1
N3:
IF I1 < L1 THEN
IF C1$(K, I1) = C1$(K, L1) THEN
C1$(K, I1) = " ": LOCATE 12, Q1: PRINT
C1$(K, I1)
ELSE
LOCATE 12, Q1
PRINT C1$(K, I1)
END IF
L1 = L1 - 1
IF L1 > 1 THEN
GOTO N3
ELSE
GOTO S3
END IF
ELSEIF I1 = L1 THEN
L1 = L1 - 1
IF L1 > 1 THEN
GOTO N3
ELSE
GOTO S3
END IF
S3:
Q1 = Q1 + 1
I1 = I1 + 1
L1 = A1
IF I1 >= A1 THEN
GOTO L3
ELSE
GOTO N3
END IF
L3:
K11(K) = 0
FOR I = 30 TO 30 + A1
B1$ = CHR$(SCREEN(12, I))
IF B1$ < " " THEN
K11(K) = K11(K) + 1
C11$(K, K11(K)) = B1$
ELSE
GOTO MM3
END IF
MM3:
NEXT I
UZ1 = LEN(UFG2$(K))
A1 = 0
FOR I = 40 TO UZ1 + 40

```

```

B1$ = CHR$(SCREEN(6, I))
IF B1$ < "(" AND B1$ < ")" AND B1$ <
"X" AND B1$ < "X" AND B1$ < "." AND
B1$ < "+" AND B1$ < "*" AND B1$ <
"/" AND B1$ < "^" AND B1$ < "1" AND
B1$ < "2" AND B1$ < "3" AND B1$ <
"4" AND B1$ < "5" AND B1$ < "6" AND
B1$ < "7" AND B1$ < "8" AND B1$ <
"9" THEN
A1 = A1 + 1
C2$(K, A1) = B1$
ELSE
GOTO M4
END IF
M4:
NEXT I
Q1 = 30: L2 = A1
I2 = 1
N4:
IF I2 < L2 THEN
IF C2$(K, I2) = C2$(K, L2) THEN
C2$(K, I2) = " ": LOCATE 13, Q1: PRINT
C2$(K, I2)
ELSE
LOCATE 13, Q1: PRINT C2$(K, I2)
END IF
L2 = L2 - 1
IF L2 > 1 THEN
GOTO N4
ELSE
GOTO S4
END IF
ELSEIF I2 = L2 THEN
L2 = L2 - 1
IF L2 > 1 THEN
GOTO N4
ELSE
GOTO S4
END IF
END IF
S4:
Q1 = Q1 + 1
I2 = I2 + 1
L2 = A1
IF I2 >= A1 THEN
GOTO L4
ELSE
GOTO N4
END IF
L4:
K12(K) = 0
FOR I = 30 TO 30 + A1
B1$ = CHR$(SCREEN(13, I))
IF B1$ < " " THEN
K12(K) = K12(K) + 1
C12$(K, K12(K)) = B1$

```

```

ELSE
GOTO MM4
END IF
MM4:
NEXT I
UZ2 = LEN(UFG3$(K))
A2 = 0
FOR I = 40 TO UZ2 + 40
B2$ = CHR$(SCREEN(8, I))
IF B2$ < "(" AND B2$ < ")" AND B2$ <
"X" AND B2$ < "X" AND B2$ < "." AND
B2$ < "+" AND B2$ < "*" AND B2$ <
"/" AND B2$ < "^" AND B2$ < "1" AND
B2$ < "2" AND B2$ < "3" AND B2$ <
"4" AND B2$ < "5" AND B2$ < "6" AND
B2$ < "7" AND B2$ < "8" AND B2$ <
"9" THEN
A2 = A2 + 1
C3$(K, A2) = B2$
ELSE
GOTO M5
END IF
M5:
NEXT I
Q2 = 30: L3 = A2
I3 = 1
N5:
IF I3 < L3 THEN
IF C3$(K, I3) = C3$(K, L3) THEN
C3$(K, I3) = " ": LOCATE 14, Q2: PRINT
C3$(K, I3)
ELSE
LOCATE 14, Q2
PRINT C3$(K, I3)
END IF
L3 = L3 - 1
IF L3 > 1 THEN
GOTO N5
ELSE
GOTO S5
END IF
ELSEIF I3 = L3 THEN
L3 = L3 - 1
IF L3 > 1 THEN
GOTO N5
ELSE
GOTO S5
END IF
END IF
S5:
Q2 = Q2 + 1
I3 = I3 + 1
L3 = A2
IF I3 >= A2 THEN
GOTO L5
ELSE

```

```

GOTO N5
END IF
L5:
K13(K) = 0
FOR I = 30 TO 30 + A1
B1$ = CHR$(SCREEN(14, I))
IF B1$ < " " THEN
K13(K) = K13(K) + 1
C13$(K, K13(K)) = B1$
ELSE
GOTO MM5
END IF
MM5:
NEXT I
LOCATE 18, 15: PRINT "GİRİLENLER
DOĞRU MU.....(E/H)": QINPUT 50, 18, Y$,
A, 1, KOD$
LOOP UNTIL Y$ < "H" OR Y$ < "H"

WRITE #1, UFG1$(K), UFG2$(K),
UFG3$(K)
FOR I = 1 TO K11(K)
WRITE #4, C11$(K, I)
NEXT I
FOR I = 1 TO K12(K)
WRITE #4, C12$(K, I)
NEXT I
FOR I = 1 TO K13(K)
WRITE #4, C13$(K, I)
NEXT I
WRITE #5, K11(K), K12(K), K13(K)
NEXT K
WRITE #1, UFC$
KAPAT1:
CLOSE #1, #4, #5
CLS 2
CASE 2
OPEN "O", #2, "UKKAYIT.DAT"
OPEN "O", #3, "SAYI.DAT"
TEKRAR:
CLS 2
COLOR 14, 7
FOR A = 1 TO 10
LOCATE 3, 9: COLOR 12
PRINT *****GİRİS İCİN SECİLECEK
UYELİK KUMELERİ*****
LOCATE 4 + A, 8: COLOR 14 PRINT A;
".GİRİS İCİN KAC AYRI UYELİK KUMESİ
TANIMLAYACAKSINIZ...";
QINPUT 66, 4 + A, A$, GUK(A), 1, KOD$
NEXT A
CLS 2
S = 5
FOR A = 1 TO 10
FOR I = 1 TO GUK(A)

```

```

LOCATE S, 15: PRINT A; ". "; "GİRİS İCİN
"; I; ". "; "UYELİK KUMESİ...";
QINPUT 51, S, GUK$(A, I), JHG, 3, KOD$
IF S = 17 THEN
S = 5
CLS 2
END IF
S = S + 1
NEXT I: NEXT A
UYT:
LOCATE 19, 14: PRINT "GİRİLENLER
DOĞRU MU (E/H)..."
QINPUT 44, 19, İ$, JHG, 1, KOD$
IF İ$ = "E" OR İ$ = "E" THEN
FOR A = 1 TO 10: FOR I = 1 TO GUK(A):
WRITE #2, GUK$(A, I): NEXT I: NEXT A
ELSEIF İ$ = "H" OR İ$ = "H" THEN GOTO
TEKRAR
ELSE GOTO UYT
END IF
DO
CLS 2
LOCATE 4, 5: COLOR 12
PRINT *****CIKIS İCİN SECİLECEK
UYELİK KUMELERİ*****
LOCATE 6, 5: COLOR 14
PRINT "CIKIS İCİN KAC AYRI UYELİK
KUMESİ TANIMLAYACAKSINIZ...": QIN-
PUT 60, 6, A$, CUK, 1, KOD$
CLS 2
LOCATE 4, 5: COLOR 12
PRINT *****CIKIS İCİN SECİLECEK
UYELİK KUMELERİ*****
S = 7
FOR I = 1 TO CUK
LOCATE S, 15: COLOR 14 PRINT "CIKIS
İCİN "; I; ". UYELİK KUMESİ...";
QINPUT 48, S, CUK$(I), A, 2, KOD$
S = S + 1
NEXT I
LOCATE 15, 15: COLOR 14 PRINT
"GİRİLENLER DOĞRU MU (E/H)..."
QINPUT 44, 15, İ$, JHG, 1, KOD$
LOOP UNTIL İ$ = "E" OR
İ$ = "E"
FOR I = 1 TO CUK
WRITE #2, CUK$(I): NEXT I
FOR I = 1 TO 10: WRITE #3, GUK(I):
NEXT I
WRITE #3, CUK
CLOSE #2, #3
COLOR , 7
CLS 2
CASE 3
CALL OKU1
END SELECT

```

```

END SUB
SUB STDMENU
CLS 2
LOCATE 3, 3: PRINT STRING$(80,
CHR$(205))
LOCATE 8, 3: PRINT STRING$(80,
CHR$(205))
FOR I = 4 TO 8: LOCATE I, 1: PRINT
CHR$(186)
LOCATE I, 80: PRINT CHR$(186): NEXT I
FOR I = 4 TO 8: LOCATE I, 3: COLOR 8, 7:
PRINT STRING$(78, CHR$(32)): NEXT I
END SUB
SUB TANIM
SCREEN 9
COLOR , 7
ESC$ = CHR$(27)
BACKSPACE$ = CHR$(8)
INSERT$ = CHR$(0) + CHR$(82)
UP$ = CHR$(0) + CHR$(72)
DOWN$ = CHR$(0) + CHR$(80)
LLEFT$ = CHR$(0) + CHR$(75)
RRIGHT$ = CHR$(0) + CHR$(77)
SPCE$ = CHR$(32)
ENTER$ = CHR$(13)
F1$ = CHR$(0) + CHR$(59)
F2$ = CHR$(0) + CHR$(60)
F3$ = CHR$(0) + CHR$(61)
F4$ = CHR$(0) + CHR$(62)
F5$ = CHR$(0) + CHR$(63)
F6$ = CHR$(0) + CHR$(64)
F7$ = CHR$(0) + CHR$(65)
F8$ = CHR$(0) + CHR$(66)
F9$ = CHR$(0) + CHR$(67)
F10$ = CHR$(0) + CHR$(68)
END SUB
SUB TUS
DO: IS = INKEY$
LOOP UNTIL IS <> ""
END SUB

```

***** SEÇİM2*****

"Bu bölümde; üyelik kümelerine ait sınır değerleri ve kural tabanı tanımlanmaktadır."

```

DECLARE SUB QINPUT (X!, Y!, STRNG$,
NUMERIC!, DIMENSION!, KODE$)
DECLARE SUB SEC2 () DECLARE SUB
TANIM ()
DECLARE SUB QINPUT (X!, Y!, STRNG$,
NUMERIC!)
DECLARE SUB GINPUT (DIMENSION!,
KODE$): DECLARE SUB SEC2 ()
$INCLUDE: 'QBX.Bİ'

```

```

COMMON SHARED KATAR$, ESC$,
BACKSPACE$, INSERT$,
COMMON SHARED UP$, DOWN$,
LLEFT$, RRIGHT$, SPCE$, ENTER$, F1$
COMMON SHARED F2$, F3$, F4$, F5$,
F6$, F7$, F8$, F9$, F10$
DIM SHARED INREGS AS REGTYPE,
OUTREGS AS REGTYPE
INREGS.AX = &H1: CALL INTER-
RUPT(&H33, INREGS, OUTREGS)
INREGS.AX = &H4: INREGS.CX = 632
INREGS.DX = 192: CALL INTER-
RUPT(&H33, INREGS, OUTREGS)
INREGS.AX = &H3: SEC2
LOCATE 22, 30
PRINT "HER HANGİ BİR TUSA BASINIZ"
DO: IS = INKEY$
LOOP UNTIL IS <> ""
CLEAR
OPEN "O", #1, "PARAM.FLM"
WRITE #1, "1"
CLOSE : RUN "36A.BAS"
SUB QINPUT (X, Y, STRNG$, NUMERIC,
DIMENSION, KODE$)
KPKL2 = Y
KPKL4 = X
COLOR 15
TANIM
STRNG$ = ""
NUMERIC = 0
KOD$ = ""
DO
ASS = INKEY$
K = K + 1
IF K = 5 THEN LOCATE KPKL2, KPKL4:
PRINT STRNG$, "Ü"
IF K = 10 THEN LOCATE KPKL2, KPKL4:
PRINT STRNG$, " ": K = 0
IF ASS = "" THEN GOTO DON
SELECT CASE ASS
CASE ESC$
KOD$ = ESC$
GOTO DONSON
CASE BACKSPACE$
ASİ = LEN(STRNG$)
IF ASİ = 0 THEN BEEP
GOTO DON
LOCATE KPKL2, KPKL4
PRINT SPACE$(ASİ + 1)
STRNG$ = LEFT$(STRNG$, ASİ - 1)
NUMERIC = VAL(STRNG$)
LOCATE KPKL2, KPKL4
PRINT STRNG$ + "Ü"
LOCATE KPKL2, KPKL4
PRINT STRNG$, " "
CASE UP$

```

```

KOD$ = ASS
GOTO DONSON
CASE DOWN$
KODE$ = ASS
GOTO DONSON
CASE LLEFT$
KOD$ = ASS
GOTO DONSON
CASE RRRIGHT$
KOD$ = ASS
GOTO DONSON
CASE SPCE$
CASE ENTER$
KOD$ = ENTER$
GOTO DONSON
CASE F1$
KOD$ = ASS
GOTO DONSON
CASE F2$
KOD$ = ASS
GOTO DONSON
CASE F3$
KOD$ = ASS
GOTO DONSON
CASE F4$
KOD$ = ASS
GOTO DONSON
CASE F5$
KOD$ = ASS
GOTO DONSON
CASE F6$
KOD$ = ASS
GOTO DONSON
CASE F7$
KOD$ = ASS
GOTO DONSON
CASE F8$
KOD$ = ASS
GOTO DONSON
CASE F9$
KOD$ = ASS
GOTO DONSON
CASE F10$
KOD$ = ASS
GOTO DONSON
CASE ELSE
IF LEN(STRNG$) = DIMENSION AND ASS
< > "" THEN BEEP
GOTO DON
IF ASS < CHR$(32) THEN GOTO DON
STRNG$ = STRNG$ + ASS
LOCATE KPKL2, KPKL4: PRINT STRNG$;
"U"
LOCATE KPKL2, KPKL4: PRINT STRNG$;
" "
NUMERIC = VAL(STRNG$)

```

```

END SELECT
DON:
LOOP UNTIL KODE$ = ESC$
DONSON:
LOCATE KPKL2, KPKL4
PRINT STRNG$; " "
END SUB
SUB SEC2
OPEN "I", #8, "NUMBER.DAT"
INPUT #8, X
CLOSE #8
REDIM ARA1(10, 1 TO 7, 2)
REDIM ARA2(10, 1 TO 7, 2)
REDIM ARA3(10, 1 TO 7, 2)
REDIM GUK(10)
Z = X - 3
SELECT CASE Z
CASE 1
COLOR 14, 7
SCREEN 9, 0
COLOR , 7
OPEN "O", #7, "SINIR.DAT"
OPEN "I", #3, "SAYI.DAT"
FOR I = 1 TO 10
INPUT #3, GUK(I)
NEXT I
INPUT #3, CUK
CLOSE #3
SATIR = 5: L = 5
FOR A = 1 TO 10
FOR I = 1 TO GUK(A)
FOR KL = 1 TO 2
LOCATE SATIR, L: COLOR 15, 7: PRINT
A; ". GIRIS"; I; ". UYELIK KUMESINDE
1.ARALIK ICIN"; KL; ". SINIR SABITL.";
LOCATE SATIR, 67: COLOR 15, 7: PRINT
"
QINPUT 67, SATIR, A$, ARA1(A, I, KL), 5,
KOD$
IF SATIR = 19 THEN
DO: I$ = INKEY$
LOOP UNTIL I$ = "D" OR I$ = "D"
SATIR = 5
VIEW PRINT 3 TO 23
VIEW PRINT 1 TO 23
ELSE
GOTO ML7
END IF
ML7:
SATIR = SATIR + 1
NEXT KL
FOR KL = 1 TO 2
LOCATE SATIR, L: COLOR 15, 7: PRINT
A; ". GIRIS"; I; ". UYELIK KUMESINDE
2.ARALIK ICIN"; KL; ". SINIR SABITL.";

```

```

LOCATE SATIR, 67: COLOR 15, 7: PRINT
"
QINPUT 67, SATIR, A$, ARA2(A, I, KL), 5,
KOD$
IF SATIR = 19 THEN
DO: I$ = INKEY$: LOOP UNTIL I$ = "D"
OR I$ = "D"
SATIR = 5
VIEW PRINT 3 TO 23
VIEW PRINT 1 TO 23
ELSE
GOTO ML8
END IF
ML8:
SATIR = SATIR + 1
NEXT KL
FOR KL = 1 TO 2
QINPUT 67, SATIR, A$, ARA3(A, I, KL), 5,
KOD$
IF SATIR = 19 THEN
DO: I$ = INKEY$: LOOP UNTIL I$ = "D"
OR I$ = "D"
SATIR = 5
VIEW PRINT 3 TO 23
VIEW PRINT 1 TO 23
ELSE
GOTO ML9
END IF
ML9:
SATIR = SATIR + 1
NEXT KL
NEXT I
NEXT A
FOR A = 1 TO 10
FOR I = 1 TO GUK(A)
FOR KL = 1 TO 2
WRITE #7, ARA1(A, I, KL)
NEXT KL
FOR KL = 1 TO 2
WRITE #7, ARA2(A, I, KL)
NEXT KL
FOR KL = 1 TO 2
WRITE #7, ARA3(A, I, KL)
NEXT KL
NEXT I
NEXT A
CLOSE #7
CASE 2
RUN "MATRISG6.BAS"
CASE 3
F = 22: LOCATE 12, 30
PRINT F
END SELECT
END SUB
SUB TANIM
SCREEN 9

```

```

COLOR , 7
ESC$ = CHR$(27)
BACKSPACES$ = CHR$(8)
INSERT$ = CHR$(0) + CHR$(82)
UP$ = CHR$(0) + CHR$(72)
DOWN$ = CHR$(0) + CHR$(80)
LLEFT$ = CHR$(0) + CHR$(75) RRRIGHT$
= CHR$(0) + CHR$(77)
SPCE$ = CHR$(32)
ENTER$ = CHR$(13)
F1$ = CHR$(0) + CHR$(59)
F2$ = CHR$(0) + CHR$(60)
F3$ = CHR$(0) + CHR$(61)
F4$ = CHR$(0) + CHR$(62)
F5$ = CHR$(0) + CHR$(63)
F6$ = CHR$(0) + CHR$(64)
F7$ = CHR$(0) + CHR$(65)
F8$ = CHR$(0) + CHR$(66)
F9$ = CHR$(0) + CHR$(67)
F10$ = CHR$(0) + CHR$(68)
END SUB

```

*****MATRISG6*****

"Bu bölümde kural tabanı tanımlanmaktadır."

```

DECLARE SUB QINPUT (KPKL4!,
KPKL2!, STRNG$, NUMERIC!, DIMENSION!,
KODE$)
DECLARE SUB QINPUT (KPKL4!,
KPKL2!, STRNG$)
DECLARE SUB GINPUT (NUMERIC!,
DIMENSION!, KODE$)
DECLARE SUB QINPUT (KPKL4!,
KPKL2!, STRNG$,
NUMERIC!, DIMENSION!, KODE$)
DECLARE SUB GRAPHMATRIS ()
DECLARE SUB KUTUK () DECLARE SUB
TANIM ()
DECLARE SUB QINPUT (X!, Y!, STRNG$,
NUMERIC!, DIMENSION!, KODE$)
DECLARE SUB GRAPHMATRIS
COMMON SHARED KATAR$, CUK, ESC$,
BACKSPACES$, INSERT$
COMMON SHARED UP$, DOWN$,
LLEFT$, RRRIGHT$, SPCE$, ENTER$, F1$,
F2$
COMMON SHARED F3$, F4$, F5$, F6$,
F7$, F8$, F9$, F10$
DIM SHARED INREGS AS REGTYPE,
OUTREGS AS REGTYPE
INREGS.AX = &H1: CALL INTER-
RUPT(&H33, INREGS, OUTREGS)
REDIM SHARED GUK(10)
REDIM SHARED GUK$(10, 7): REDIM
SHARED CUK$(7)

```

```

SCREEN 12
GRAPHMATRİS
CLEAR
OPEN "O", #1, "PARAM.FLM" WRITE #1,
"1"
CLOSE
OPEN "O", 1, "GRAPHMAK.FLM"
WRITE #1, A, B, C, D, E
CLOSE
RUN "36A.BAS"
SUB GRAPHMATRİS
T = 1
T2 = 2
KUL = 5
KU = 1
KUTUK
OPEN "O", #12, "KURALL.DAT"
HUY:
REDIM KURAL$(KU TO KUL, 10, 49)
YUT:
T1 = T2
X = GUK(T)
YUT1:
Y = GUK(T + 1)
PAINT (0, 0), 3
LINE (100, 96)-(550, 400), 8, BF
LINE (100, 96)-(550, 400), 8, B
LINE (0, 0)-(100, 96), 8
LINE (0, 480)-(100, 400), 8
LINE (639, 0)-(550, 100), 8
LINE (639, 480)-(550, 400), 8
VIEW (100, 96)-(550, 400), 0, 15
LOCATE 2, 7: PRINT " "
LOCATE 3, 7: PRINT " "
LOCATE 2, 7
COLOR 14
PRINT T1, " "
LOCATE 3, 7
COLOR 14
PRINT "GİRİS"
LOCATE 4, 2
PRINT " "
LOCATE 5, 2
PRINT " "
LOCATE 4,
COLOR 14
PRINT T, " "
LOCATE 5, 2
COLOR 14
PRINT "GİRİS"
LOCATE 10, 71
COLOR 12
PRINT "ÇIKIŞ U.K"
FOR İ = 11 TO 18
LOCATE İ, 71: PRINT " "
NEXT İ

```

```

Kİ = 11
FOR İ = 1 TO CUK
LOCATE Kİ, 74
COLOR 14
PRINT CUK$(İ)
Kİ = Kİ + 1
NEXT İ
IF Y = 1 THEN
KPKL = 16
KPKL2 = 16
SFR = 0
ELSEIF Y = 2 THEN
KPKL = 12
KPKL2 = 12
SFR = 9
ELSEIF Y = 3 THEN
KPKL = 10
KPKL2 = 10
SFR = 6
ELSEIF Y = 4 THEN
KPKL = 9
KPKL2 = 9
SFR = 5
ELSEIF Y = 5 THEN
KPKL = 8
KPKL2 = 8
SFR = 4
ELSEIF Y = 6 THEN
KPKL = 9
KPKL2 = 9
SFR = 3
ELSE
KPKL = 7
KPKL2 = 7
SFR = 3
END IF
REM KPKL1...: BASLANGİC SUTUN SA-
YİSİ
REM SFR1...: SUTUN ARTIRMA SAYISI
IF X = 1 THEN
KPKL1 = 40
KPKL4 = 40
SFR1 = 0
ELSEIF X = 2 THEN
KPKL1 = 26
KPKL4 = 26
SFR1 = 26
ELSEIF X = 3 THEN
KPKL1 = 22
KPKL4 = 22
SFR1 = 18
ELSEIF X = 4 THEN
KPKL1 = 19
KPKL4 = 19
SFR1 = 14
ELSEIF X = 5 THEN

```

```

KPKL1 = 18
KPKL4 = 18
SFR1 = 11
ELSEIF X = 6 THEN
KPKL1 = 17
KPKL4 = 17
SFR1 = 9
ELSE
KPKL1 = 15
KPKL4 = 15
SFR1 = 8
END IF
IF Y > 1 THEN
CONS1 = ((Y - 1) * SFR + KPKL2)
ELSE
CONS1 = KPKL2
END IF
IF X > 1 THEN
CONS2 = ((X - 1) * SFR1 + KPKL4)
ELSE
CONS2 = KPKL4
END IF
ARALIKX = X * 8 * 100
ARALIKY = Y * 16 * 100
X1 = CINT(ARALIKX / X)
Y1 = CINT(ARALIKY / Y)
WINDOW SCREEN (0, 0)-(ARALIKX, A-
RALIKY)
FOR I = 1 TO ARALIKX STEP X1
LINE (I, 0)-(I, ARALIKY), 15
NEXT
FOR I = 1 TO ARALIKY STEP Y1
LINE (0, I)-(ARALIKX, I), 15
NEXT
TANIM
G:
FOR I = 1 TO Y
LOCATE KPKL, 6: PRINT " "
LOCATE KPKL, 6: COLOR 15 PRINT
GUK$(T, I)
KPKL = KPKL + SFR
NEXT I
FOR I = 1 TO X
LOCATE KPKL3, KPKL1
PRINT " "
LOCATE KPKL3, KPKL1
PRINT GUK$(T1, I)
KPKL1 = KPKL1 + SFR1
NEXT I
IF X > 1 AND Y = 1 THEN GOTO HJ12
IF X = 1 AND Y > 1 THEN GOTO HJ13
IF X > 1 AND Y > 1 THEN GOTO HJ11
IF X = 1 AND Y = 1 THEN GOTO HJ14
HJ14:
T3 = 0
FOR L = KPKL2 TO CONS1

```

```

FOR I = KPKL4 TO CONS2
T3 = T3 + 1
QINPUT I, L, KURAL$(T, T1, T3), KJH, 3,
KOD$
WRITE #12, KURAL$(T, T1, T3)
NEXT I
NEXT L
GOTO MMMMM
HJ13:
T3 = 0
FOR L = KPKL2 TO CONS1 STEP SFR
FOR I = KPKL4 TO CONS2
T3 = T3 + 1
QINPUT I, L, KURAL$(T, T1, T3), KJH, 3,
KOD$
WRITE #12, KURAL$(T, T1, T3)
NEXT I
NEXT L
GOTO MMMMM
HJ12:
T3 = 0
FOR L = KPKL2 TO CONS1
FOR I = KPKL4 TO CONS2 STEP SFR1
T3 = T3 + 1
QINPUT I, L, KURAL$(T, T1, T3), KJH, 3,
KOD$
WRITE #12, KURAL$(T, T1, T3)
NEXT I
NEXT L
GOTO MMMMM
HJ11:
T3 = 0
FOR L = KPKL2 TO CONS1 STEP SFR
FOR I = KPKL4 TO CONS2 STEP SFR1
T3 = T3 + 1
QINPUT I, L, KURAL$(T, T1, T3), KJH, 3,
KOD$
WRITE #12, KURAL$(T, T1, T3)
NEXT I
NEXT L
MMMMM:
T1 = T1 + 1
IF T1 <= 10 THEN
GOTO YUT1
ELSE
T = T + 1
GOSUB YUT2
T2 = T2 + 1
GOTO YUT
END IF
YUT2:
IF T = 6 THEN
ERASE KURAL$
KUL = 9
KU = 6
T2 = T2 + 1

```



```

GOTO HUY
ELSEIF T <= 9 THEN
RETURN
ELSE
GOTO KTY
END IF
KTY:
CLOSE #12
END SUB
SUB KUTUK
OPEN "I", #2, "UKKAYIT.DAT"
OPEN "I", #3, "SAYI.DAT"
FOR I = 1 TO 10
INPUT #3, GUK(I)
FOR K = 1 TO GUK(I)
INPUT #2, GUK$(I, K)
NEXT K
NEXT I
INPUT #3, CUK
FOR H = 1 TO CUK
INPUT #2, CUK$(H)
NEXT H
CLOSE #2, #3
END SUB
SUB QINPUT (KPKL4, KPKL2, STRNG$,
NUMERIC, DIMENSION, KODE$)
COLOR 15
TANIM
STRNG$ = ""
NUMERIC = 0
KOD$ = ""
DO
ASS$ = INKEY$
SELECT CASE ASS$
CASE ESC$
KOD$ = ESC$
GOTO DONSON
CASE BACKSPACES
ASI = LEN(STRNG$)
IF ASI = 0 THEN BEEP: GOTO DON
LOCATE KPKL2, KPKL4: PRINT SPA-
CES(ASI + 1)
STRNG$ = LEFT$(STRNG$, ASI - 1)
NUMERIC = VAL(STRNG$)
LOCATE KPKL2, KPKL4: PRINT STRNG$
+ "Ü"
CASE UP$
KOD$ = ASS$
GOTO DONSON
CASE DOWN$
KODE$ = ASS$
GOTO DONSON
CASE LLEFT$
KOD$ = ASS$
GOTO DONSON
CASE RRIGHT$

```

```

KOD$ = ASS$
GOTO DONSON
CASE SPCE$
CASE ENTER$
KOD$ = ENTER$
GOTO DONSON
CASE F1$
KOD$ = ASS$
GOTO DONSON
CASE F2$
KOD$ = ASS$
GOTO DONSON
CASE F3$
KOD$ = ASS$
GOTO DONSON
CASE F4$
KOD$ = ASS$
GOTO DONSON
CASE F5$
KOD$ = ASS$
GOTO DONSON
CASE F6$
KOD$ = ASS$
GOTO DONSON
CASE F7$
KOD$ = ASS$
GOTO DONSON
CASE F8$
KOD$ = ASS$
GOTO DONSON
CASE F9$
KOD$ = ASS$
GOTO DONSON
CASE F10$
KOD$ = ASS$
GOTO DONSON
CASE ELSE
IF LEN(STRNG$) = DIMENSION AND ASS$
<> "" THEN BEEP: GOTO DON
IF ASS$ < CHR$(32) THEN GOTO DON
STRNG$ = STRNG$ + ASS$
LOCATE KPKL2, KPKL4: PRINT STRNG$,
"Ü"
NUMERIC = VAL(STRNG$)
END SELECT
DON:
LOOP UNTIL KODE$ = ESC$
DONSON:
LOCATE , , 1
END SUB
SUB TANIM STATIC
ESC$ = CHR$(27)
BACKSPACES$ = CHR$(8)
INSERT$ = CHR$(0) + CHR$(82)
UP$ = CHR$(0) + CHR$(72)
DOWN$ = CHR$(0) + CHR$(80)

```

```

LEFT$ = CHR$(0) + CHR$(75)
RIGHT$ = CHR$(0) + CHR$(77)
SPACE$ = CHR$(32)
ENTER$ = CHR$(13)
F1$ = CHR$(0) + CHR$(59)
F2$ = CHR$(0) + CHR$(60)
F3$ = CHR$(0) + CHR$(61)
F4$ = CHR$(0) + CHR$(62)
F5$ = CHR$(0) + CHR$(63)
F6$ = CHR$(0) + CHR$(64)
F7$ = CHR$(0) + CHR$(65)
F8$ = CHR$(0) + CHR$(66)
F9$ = CHR$(0) + CHR$(67)
F10$ = CHR$(0) + CHR$(68)
END SUB

```

*****ÇÖZÜM12*****

"Bu bölümde; tanımlanan sistem üzerinde bulanık mantık kuralları işletilerek sistemin kontrolü yapılmaktadır."

```

DECLARE SUB PORT () DECLARE SUB
HESAPT ()
DECLARE SUB EKSIİ () DECLARE SUB
FONSVEHES ()
DECLARE SUB OKU11 () DECLARE SUB
KUMESEC ()
$INCLUDE: 'QBX.BI'
COMMON SHARED CUK, LL, X1, X2, X3,
X4, X5, X6, X7
COMMON SHARED CUK, X8, X9, X10,
ISARET11, ISARET12, ISARET3
COMMON SHARED CUK, SAYI, SAYI1,
SAYID, ISARET, SONUC2
DIM SHARED X(10) AS INTEGER
DIM SHARED INREGS AS REGTYPE,
OUTREGS AS REGTYPE
INREGS.AX = &H1
CALL INTERRUPT(&H33, INREGS, OUT-
REGS)
PEN ON 'ENABLE LIGHT PEN.
INREGS.AX = &HD
CALL INTERRUPT(&H33, INREGS, OUT-
REGS)
TIMER ON
ON TIMER(1) GOSUB PENREPORT
DIM SHARED A1$(1 TO 30)
DIM SHARED SONUC(1 TO 30)
DIM SHARED A2$(1 TO 30)
DIM SHARED A3$(1 TO 30)
DIM SHARED A4$(1 TO 30)
DIM SHARED ARAL(10, 1 TO 7, 3)
DIM SHARED SONUC3(10, 1 TO 7)
CLOSE
PORT

```

```

KUMESEC
FONSVEHES
CLEAR
CHAIN "KURALS3"
PENREPORT AD$ = INKEY$
IF AD$ = CHR$(27) THEN END
IF AD$ = CHR$(0) + ","
THEN Y = 135
X = 68
GOTO A
IF AD$ = CHR$(0) + "<"
THEN Y = 135
X = 141
GOTO A
IF AD$ = CHR$(0) + "="
THEN Y = 135
X = 213
GOTO A
IF AD$ = CHR$(0) + ">"
THEN Y = 135
X = 285
GOTO A
IF -1 = PEN(3) THEN
AND GET X AND Y POSITION.
X = PEN(4)
Y = PEN(5)
A:
IF Y >= 132 AND Y <= 155 THEN
IF X >= 67 AND X <= 135 THEN PEN OFF
INREGS.AX = &HE
CALL INTERRUPT(&H33, INREGS, OUT-
REGS)
CLS
SCREEN 0
END
IF X >= 140 AND X <= 206 THEN PEN OFF
INREGS.AX = &HE
CALL INTERRUPT(&H33, INREGS, OUT-
REGS)
RUN "36A"
IF X >= 212 AND X <= 279 THEN
DO
K = 1
IF PEN(0) = -1 THEN
X = PEN(4)
Y = PEN(5)
IF Y >= 132 AND Y <= 155 THEN
IF X >= 67 AND X <= 135 THEN PEN OFF
INREGS.AX = &HE
CALL INTERRUPT(&H33, INREGS, OUT-
REGS)
CLS
SCREEN 0
END
IF X >= 140 AND X <= 206 THEN PEN OFF
INREGS.AX = &HE

```

```

CALL INTERRUPT(&H33, INREGS, OUT-
REGS)
RUN "36A"
IF X >= 284 AND X <= 352 THEN K = 0
END IF
END IF LOOP UNTIL K = 0
END IF
IF X >= 284 AND X <= 352 THEN K = 0
END IF
END IF
RETURN
SUB EKSII
ON PEN GOSUB PENREPORT
ISARET12 = LL + 1
TEKI:
IF A3$(ISARET12) <> ")" THEN
A3$(LL) = A3$(ISARET12)
A3$(ISARET12) = ""
LL = LL + 1
ISARET12 = ISARET12 + 1
GOTO TEKİ
ELSE
A3$(LL) = A3$(ISARET12)
A3$(ISARET12) = ""
ISARET = ISARET - 1
END IF
END SUB
SUB FONSVEHES
ON PEN GOSUB PENREPORT
REDIM GUK(10)
REDIM SAC(1, 1 TO 7)
REDIM SA(10, 1 TO 7, 2)
REDIM SA1(10, 1 TO 7, 2)
REDIM SA2(10, 1 TO 7, 2)
REDIM CUK$(1 TO 7)
REDIM C11$(10, 2)
REDIM C12$(10, 2)
REDIM C13$(10, 2)
REDIM K11(10)
REDIM K12(10)
REDIM K13(10)
REDIM UFG1$(10)
REDIM UFG2$(10)
REDIM UFG3$(10)
OPEN "I", #1, "UFKAYIT.DAT"
OPEN "I", #3, "SAYI.DAT"
OPEN "I", #4, "DEGER.DAT"
OPEN "I", #5, "DEGER1.DAT"
OPEN "I", #6, "SABITGC.DAT"
FOR H = 1 TO 10
INPUT #1, UFG1$(H), UFG2$(H),
UFG3$(H)
INPUT #5, K11(H), K12(H), K13(H)
NEXT H
FOR H = 1 TO 10
FOR I = 1 TO K11(H)
INPUT #4, C11$(H, I)
NEXT I
FOR I = 1 TO K12(H)
INPUT #4, C12$(H, I)
NEXT I
FOR I = 1 TO K13(H)
INPUT #4, C13$(H, I)
NEXT I
NEXT H
CLOSE #1, #4, #5
FOR I = 1 TO 10
INPUT #3, GUK(I)
NEXT I
INPUT #3, CUK
FOR A = 1 TO 10
FOR I = 1 TO GUK(A)
FOR KL = 1 TO K11(A)
INPUT #6, SA(A, I, KL)
NEXT KL
FOR KL = 1 TO K12(A)
INPUT #6, SA1(A, I, KL)
NEXT KL
FOR KL = 1 TO K13(A)
INPUT #6, SA2(A, I, KL)
NEXT KL
NEXT I
NEXT A
FOR I = 1 TO CUK
INPUT #6, SAC(1, I)
NEXT I
CLOSE #3, #6
SAYII = 1
KONT:
SELECT CASE SAYII
CASE 1
SND = 0
FOR M = 1 TO GUK(1)
IF ARAL(1, M, 1) = 1 THEN
GOTO ATANAN1
ELSE
GOTO SORAN1
END IF
SORAN1:
IF ARAL(1, M, 2) = 1 THEN
GOTO ATANAN2
ELSE
GOTO SORAN2
END IF
SORAN2:
IF ARAL(1, M, 3) = 1 THEN
GOTO ATANAN3
ELSE
GOTO DONGU
END IF
ATANAN1:
ISARET = 0

```

```

SAYID = 0
SAYID3 = 0
ISARET11 = 0
UZ = LEN(UFG1$(1))
FOR NNN = 1 TO UZ
F1$ = MID$(UFG1$(1), NNN, 1)
IF F1$ = "(" THEN
ISARET11 = 0
SAYID3 = 0
LL = NNN + 1
LM = NNN
ISARET = ISARET + 1
A3$(ISARET) = F1$
ELSEIF F1$ = ")" THEN
ISARET = ISARET + 1
A3$(ISARET) = F1$
GOTO KARSİ
ELSEIF F1$ = "*" OR F1$ = "/" OR F1$ =
"+" OR F1$ = "-" OR F1$ = "^" THEN
ISARET11 = ISARET11 + 1
ISARET = ISARET + 1
A3$(ISARET) = F1$
ELSE
SAYID3 = SAYID3 + 1
SAYID = SAYID + 1
A4$(SAYID) = F1$
END IF
DONE:
NEXT NNN
GOTO SEAN
KARSİ:
IF ISARET11 <> 0 AND SAYID3 <> 0 AND
ISARET11 = SAYID3 THEN
EKSIİ
GOTO SANİ
ELSE
GOTO DONE
END IF
SANİ:
FOR İ = 1 TO SAYID3
IF A4$(İ) = "X" THEN
A4$(İ) = STR$(-X1)
ELSE
GOTO MLKN
END IF
MLKN:
NEXT İ
GOTO DONE
SEAN:
IF SAYID <= 1 AND F1$ <> "X" THEN
SND = SND + 1
SONUC3(1, SND) = SA(1, M, 1)
ARAL(1, M, 1) = 0
GOTO DONGU
ELSE
GOTO SAN

```

```

END IF
SAN:
FOR İ = 1 TO SAYID
IF A4$(İ) = "X" THEN
A4$(İ) = STR$(X1)
ELSEIF A4$(İ) = STR$(-X1) THEN
GOTO MLK
ELSE
GOTO MLK
END IF
MLK:
NEXT İ
FOR İ = 1 TO SAYID
FOR N = 1 TO K11(1)
IF A4$(İ) = C11$(1, N) THEN
A4$(İ) = STR$(SA(1, M, N))
ELSE
GOTO FGH
END IF
FGH:
NEXT N
NEXT İ
HESAPT
SND = SND + 1
SONUC3(1, SND) = SONUC2
ARAL(1, M, 1) = 0
GOTO DONGU
ATANAN2:
ISARET = 0
SAYID = 0
SAYID3 = 0
ISARET11 = 0
UZ = LEN(UFG2$(1))
FOR NNN = 1 TO UZ
F1$ = MID$(UFG2$(1), NNN, 1)
IF F1$ = "(" THEN
ISARET11 = 0
SAYID3 = 0
LL = NNN + 1
LM = NNN
ISARET = ISARET + 1
A3$(ISARET) = F1$
ELSEIF F1$ = ")" THEN
ISARET = ISARET + 1
A3$(ISARET) = F1$
GOTO KARSİİ
ELSEIF F1$ = "*" OR F1$ = "/" OR F1$ =
"+" OR F1$ = "-" OR F1$ = "^" THEN
ISARET11 = ISARET11 + 1
ISARET = ISARET + 1
A3$(ISARET) = F1$
ELSE
SAYID3 = SAYID3 + 1
SAYID = SAYID + 1
A4$(SAYID) = F1$
END IF

```

```

DONE1:
NEXT NNN
GOTO SEAN1
KARS1:
IF ISARET11 < 0 AND SAYID3 < 0 AND
ISARET11 = SAYID3 THEN
EKS1:
GOTO SAN1
ELSE
GOTO DONE1
END IF
SAN1:
FOR I = 1 TO SAYID3
IF A4$(I) = "X" THEN
A4$(I) = STR$(-X1)
ELSE
GOTO MLKN1
END IF
MLKN1:
NEXT I
GOTO DONE1
SEAN1:
IF SAYID <= 1 AND F1$ < "X" THEN
SND = SND + 1
SONUC3(1, SND) = SA1(1, M, 1)
ARAL(1, M, 2) = 0
GOTO DONGU
ELSE
GOTO SAN1
END IF
SAN1:
FOR I = 1 TO SAYID
IF A4$(I) = "X" THEN
A4$(I) = STR$(X1)
ELSE
GOTO MLK1
END IF
MLK1:
NEXT I
FOR I = 1 TO SAYID
FOR N = 1 TO K12(1)
IF A4$(I) = C12$(1, N) THEN
A4$(I) = STR$(SA1(1, M, N))
ELSE
GOTO FGH1
END IF
FGH1:
NEXT N
NEXT I
HESAPT
SND = SND + 1
SONUC3(1, SND) = SONUC2
ARAL(1, M, 2) = 0
GOTO DONGU
ATANAN3:
ISARET = 0

```

```

SAYID = 0
SAYID3 = 0
ISARET11 = 0
UZ = LEN(UFG3$(1))
FOR NNN = 1 TO UZ
F1$ = MID$(UFG3$(1), NNN, 1)
IF F1$ = "(" THEN
ISARET11 = 0
SAYID3 = 0
LL = NNN + 1
LM = NNN
ISARET = ISARET + 1
A3$(ISARET) = F1$
ELSEIF F1$ = ")" THEN
ISARET = ISARET + 1
A3$(ISARET) = F1$
GOTO KARS2
ELSEIF F1$ = "*" OR F1$ = "/" OR F1$ =
"+" OR F1$ = "-" OR F1$ = "^" THEN
ISARET11 = ISARET11 + 1
ISARET = ISARET + 1
A3$(ISARET) = F1$
ELSE
SAYID3 = SAYID3 + 1
SAYID = SAYID + 1
A4$(SAYID) = F1$
END IF
DONE2:
NEXT NNN
GOTO SEAN2
KARS2:
IF ISARET11 < 0 AND SAYID3 < 0 AND
ISARET11 = SAYID3 THEN
EKS2:
GOTO SAN2
ELSE
GOTO DONE2
END IF
SAN2:
FOR I = 1 TO SAYID3
IF A4$(I) = "X" THEN
A4$(I) = STR$(-X1)
ELSE
GOTO MLKN2
END IF
MLKN2:
NEXT I
GOTO DONE2
SEAN2:
IF SAYID <= 1 AND F1$ < "X" THEN
SND = SND + 1
SONUC3(1, SND) = SA2(1, M, 1)
ARAL(1, M, 3) = 0
GOTO DONGU
ELSE
GOTO SAN2

```

```

END IF
SAN2:
FOR I = 1 TO SAYID
IF A4$(I) = "X" THEN
A4$(I) = STR$(X1)
ELSE
GOTO MLK2
END IF
MLK2:
NEXT I
FOR I = 1 TO SAYID
FOR N = 1 TO K13(1)
IF A4$(I) = C13$(1, N) THEN
A4$(I) = STR$(SA2(1, M, N))
ELSE
GOTO FGH2
END IF
FGH2:
NEXT N
NEXT I
HESAPT
SND = SND + 1
SONUC3(1, SND) = SONUC2
ARAL(1, M, 3) = 0
DONGU:
NEXT M
SAYI1 = 2
GOTO KONT
CASE 2
SND1 = 0
FOR M = 1 TO GUK(2)
IF ARAL(2, M, 1) = 1 THEN
GOTO ATANAN4
ELSE
GOTO SORAN3
END IF
SORAN3:
IF ARAL(2, M, 2) = 1 THEN
GOTO ATANAN5
ELSE
GOTO SORAN4
END IF
SORAN4:
IF ARAL(2, M, 3) = 1 THEN
GOTO ATANAN6
ELSE
GOTO DONGU1
END IF
ATANAN4:
ISARET = 0
SAYID = 0
SAYID3 = 0
ISARET11 = 0
UZ = LEN(UFG1$(2))
FOR NNN = 1 TO UZ
F1$ = MID$(UFG1$(2), NNN, 1)

```

```

IF F1$ = "(" THEN
ISARET11 = 0
SAYID3 = 0
LL = NNN + 1
LM = NNN
ISARET = ISARET + 1
A3$(ISARET) = F1$
ELSEIF F1$ = ")" THEN
ISARET = ISARET + 1
A3$(ISARET) = F1$
GOTO KARS13
ELSEIF F1$ = "*" OR F1$ = "/" OR F1$ =
"+" OR F1$ = "-" OR F1$ = "^" THEN
ISARET11 = ISARET11 + 1
ISARET = ISARET + 1
A3$(ISARET) = F1$
ELSE
SAYID3 = SAYID3 + 1
SAYID = SAYID + 1
A4$(SAYID) = F1$
END IF
DONE3:
NEXT NNN
GOTO SEAN3
KARS13:
IF ISARET11 < 0 AND SAYID3 < 0 AND
ISARET11 = SAYID3 THEN
EKS11
GOTO SAN13
ELSE
GOTO DONE3
END IF
SAN13:
FOR I = 1 TO SAYID3
IF A4$(I) = "X" THEN
A4$(I) = STR$(-X2)
ELSE
GOTO MLKN3
END IF
MLKN3:
NEXT I
GOTO DONE3
SEAN3:
IF SAYID <= 1 AND F1$ <> "X" THEN
SND1 = SND1 + 1
SONUC3(2, SND1) = SA(2, M, 1)
ARAL(2, M, 1) = 0
GOTO DONGU1
ELSE
GOTO SAN3
END IF
SAN3:
FOR I = 1 TO SAYID
IF A4$(I) = "X" THEN
A4$(I) = STR$(X2)
ELSEIF A4$(I) = STR$(-X2) THEN

```

```

GOTO MLK3
ELSE
GOTO MLK3
END IF
MLK3:
NEXT I
FOR I = 1 TO SAYID
FOR N = 1 TO K11(2)
IF A4$(I) = C11$(2, N) THEN
A4$(I) = STR$(SA(2, M, N))
ELSE
GOTO FGH3
END IF
FGH3:
NEXT N
NEXT I
HESAPT
SND1 = SND1 + 1
SONUC3(2, SND1) = SONUC2
ARAL(2, M, 1) = 0
GOTO DONGU1
ATANAN5:
ISARET = 0
SAYID = 0
SAYID3 = 0
ISARET11 = 0
UZ = LEN(UFG2$(2))
FOR NNN = 1 TO UZ
F1$ = MID$(UFG2$(2), NNN, 1)
IF F1$ = "(" THEN
ISARET11 = 0
SAYID3 = 0
LL = NNN + 1
LM = NNN
ISARET = ISARET + 1
A3$(ISARET) = F1$
ELSEIF F1$ = ")" THEN
ISARET = ISARET + 1
A3$(ISARET) = F1$
GOTO KARSİ4
ELSEIF F1$ = "*" OR F1$ = "/" OR F1$ =
"+" OR F1$ = "-" OR F1$ = "^" THEN
ISARET11 = ISARET11 + 1
ISARET = ISARET + 1
A3$(ISARET) = F1$
ELSE
SAYID3 = SAYID3 + 1
SAYID = SAYID + 1
A4$(SAYID) = F1$
END IF
DONE4:
NEXT NNN
GOTO SEAN4
KARSİ4:
IF ISARET11 <> 0 AND SAYID3 <> 0 AND
ISARET11 = SAYID3 THEN

```

```

EKSİİ
GOTO SANİ4
ELSE
GOTO DONE4
END IF
SANİ4:
FOR I = 1 TO SAYID3
IF A4$(I) = "X" THEN
A4$(I) = STR$(-X2)
ELSE
GOTO MLKN4
END IF
MLKN4:
NEXT I
GOTO DONE4
SEAN4:
IF SAYID <= 1 AND F1$ <> "X" THEN
SND1 = SND1 + 1
SONUC3(2, SND1) = SA1(2, M, 1)

ARAL(2, M, 2) = 0
GOTO DONGU1
ELSE
GOTO SAN4
END IF
SAN4:
FOR I = 1 TO SAYID
IF A4$(I) = "X" THEN
A4$(I) = STR$(X2)
ELSE
GOTO MLK4
END IF
MLK4:
NEXT I
FOR I = 1 TO SAYID
FOR N = 1 TO K12(2)
IF A4$(I) = C12$(2, N) THEN
A4$(I) = STR$(SA1(2, M, N))
ELSE
GOTO FGH4
END IF
FGH4:
NEXT N
NEXT I
HESAPT
SND1 = SND1 + 1
SONUC3(2, SND1) = SONUC2
ARAL(2, M, 2) = 0
GOTO DONGU1
ATANAN6:
ISARET = 0
SAYID = 0
SAYID3 = 0
ISARET11 = 0
UZ = LEN(UFG3$(2))
FOR NNN = 1 TO UZ

```

```

F1$ = MID$(UFG3$(2), NNN, 1)
IF F1$ = "(" THEN
  ISARET11 = 0
  SAYID3 = 0
  LL = NNN + 1
  LM = NNN
  ISARET = ISARET + 1
  A3$(ISARET) = F1$
ELSEIF F1$ = ")" THEN
  ISARET = ISARET + 1
  A3$(ISARET) = F1$
GOTO KARSİ5
ELSEIF F1$ = "*" OR F1$ = "/" OR F1$ =
  "+" OR F1$ = "-" OR F1$ = "^" THEN
  ISARET11 = ISARET11 + 1
  ISARET = ISARET + 1
  A3$(ISARET) = F1$
ELSE
  SAYID3 = SAYID3 + 1
  SAYID = SAYID + 1
  A4$(SAYID) = F1$
END IF
DONE5:
NEXT NNN
GOTO SEAN5
KARSİ5:
IF ISARET11 <> 0 AND SAYID3 <> 0 AND
  ISARET11 = SAYID3 THEN
  EKSİİ
GOTO SANİ5
ELSE
GOTO DONE5
END IF
SANİ5:
FOR İ = 1 TO SAYID3
  IF A4$(İ) = "X" THEN
    A4$(İ) = STR$(-X2)
  ELSE
    GOTO MLKN5
  END IF
MLKN5:
NEXT İ
GOTO DONE5
SEAN5:
IF SAYID <= 1 AND F1$ <> "X" THEN
  SND1 = SND1 + 1
  SONUC3(2, SND1) = SA2(2, M, 1)
  ARAL(2, M, 3) = 0
  GOTO DONGU1
ELSE
  GOTO SAN5
END IF
SAN5:
FOR İ = 1 TO SAYID
  IF A4$(İ) = "X" THEN
    A4$(İ) = STR$(X2)

```

```

ELSE
  GOTO MLK5
END IF
MLK5:
NEXT İ
FOR İ = 1 TO SAYID
  FOR N = 1 TO K13(2)
    IF A4$(İ) = C13$(2, N) THEN
      A4$(İ) = STR$(SA2(2, M, N))
    ELSE
      GOTO FGH5
    END IF
  FGH5:
  NEXT N
  NEXT İ
  HESAPT
  SND1 = SND1 + 1
  SONUC3(2, SND1) = SONUC2
  ARAL(2, M, 3) = 0
  DONGU1:
  NEXT M
  SAYİ1 = 3
  GOTO KONT
    CASE 3
    SND2 = 0
    FOR M = 1 TO GUK(3)
      IF ARAL(3, M, 1) = 1 THEN
        GOTO ATANAN7
      ELSE
        GOTO SORAN5
      END IF
    SORAN5:
    IF ARAL(3, M, 2) = 1 THEN
      GOTO ATANAN8
    ELSE
      GOTO SORAN6
    END IF
    SORAN6:
    IF ARAL(3, M, 3) = 1 THEN
      GOTO ATANAN9
    ELSE
      GOTO DONGU2
    END IF
    ATANAN7:
    ISARET = 0
    SAYID = 0
    SAYID3 = 0
    ISARET11 = 0
    UZ = LEN(UFG1$(3))
    FOR NNN = 1 TO UZ
      F1$ = MID$(UFG1$(3), NNN, 1)
      IF F1$ = "(" THEN
        ISARET11 = 0
        SAYID3 = 0
        LL = NNN + 1
        LM = NNN

```



```

ISARET = ISARET + 1
A3$(ISARET) = F1$
ELSEIF F1$ = ")" THEN
ISARET = ISARET + 1
A3$(ISARET) = F1$
GOTO KARSİ6
ELSEIF F1$ = "*" OR F1$ = "/" OR F1$ =
"+" OR F1$ = "-" OR F1$ = "^" THEN
ISARET11 = ISARET11 + 1
ISARET = ISARET + 1
A3$(ISARET) = F1$
ELSE
SAYİD3 = SAYİD3 + 1
SAYİD = SAYİD + 1
A4$(SAYİD) = F1$
END IF
DONE6:
NEXT NNN
GOTO SEAN6
KARSİ6:
IF ISARET11 < 0 AND SAYİD3 < 0 AND
ISARET11 = SAYİD3 THEN
EKSİİ
GOTO SANİ6
ELSE
GOTO DONE6
END IF
SANİ6:
FOR İ = 1 TO SAYİD3
IF A4$(İ) = "X" THEN
A4$(İ) = STR$(-X3)
ELSE
GOTO MLKN6
END IF
MLKN6:
NEXT İ
GOTO DONE6
SEAN6:
IF SAYİD <= 1 AND F1$ < "X" THEN
SND2 = SND2 + 1
SONUC3(3, SND2) = SA(3, M, 1)
ARAL(3, M, 1) = 0
GOTO DONGU2
ELSE
GOTO SAN6
END IF
SAN6:
FOR İ = 1 TO SAYİD
IF A4$(İ) = "X" THEN
A4$(İ) = STR$(X3)
ELSEIF A4$(İ) = STR$(-X3) THEN
GOTO MLK6
ELSE
GOTO MLK6
END IF
MLK6:

```

```

NEXT İ
FOR İ = 1 TO SAYİD
FOR N = 1 TO K11(3)
IF A4$(İ) = C11$(3, N) THEN
A4$(İ) = STR$(SA(3, M, N))
ELSE
GOTO FGH6
END IF
FGH6:
NEXT N
NEXT İ
HESAPT
SND2 = SND2 + 1
SONUC3(3, SND2) = SONUC2
ARAL(3, M, 1) = 0
GOTO DONGU2
ATANAN8:
ISARET = 0
SAYİD = 0
SAYİD3 = 0
ISARET11 = 0
UZ = LEN(UFG2$(3))
FOR NNN = 1 TO UZ
F1$ = MID$(UFG2$(3), NNN, 1)
IF F1$ = "(" THEN
ISARET11 = 0
SAYİD3 = 0
LL = NNN + 1
LM = NNN
ISARET = ISARET + 1
A3$(ISARET) = F1$
ELSEIF F1$ = ")" THEN
ISARET = ISARET + 1
A3$(ISARET) = F1$
GOTO KARSİ7
ELSEIF F1$ = "*" OR F1$ = "/" OR F1$ =
"+" OR F1$ = "-" OR F1$ = "^" THEN
ISARET11 = ISARET11 + 1
ISARET = ISARET + 1
A3$(ISARET) = F1$
ELSE
SAYİD3 = SAYİD3 + 1
SAYİD = SAYİD + 1
A4$(SAYİD) = F1$
END IF
DONE7:
NEXT NNN
GOTO SEAN7
KARSİ7:
IF ISARET11 < 0 AND SAYİD3 < 0 AND
ISARET11 = SAYİD3 THEN
EKSİİ
GOTO SANİ7
ELSE
GOTO DONE7
END IF

```

```

SAN7:
FOR I = 1 TO SAYID3
IF A4$(I) = "X" THEN
A4$(I) = STR$(-X3)
ELSE
GOTO MLKN7
END IF
MLKN7:
NEXT I
GOTO DONE7
SEAN7:
IF SAYID <= 1 AND F1$ <> "X" THEN
SND2 = SND2 + 1
SONUC3(3, SND2) = SA1(3, M, 1)
ARAL(3, M, 2) = 0
GOTO DONGU2
ELSE
GOTO SAN7
END IF
SAN7:
FOR I = 1 TO SAYID
IF A4$(I) = "X" THEN
A4$(I) = STR$(X3)
ELSE
GOTO MLK7
END IF
MLK7:
NEXT I
FOR I = 1 TO SAYID
FOR N = 1 TO K12(3)
IF A4$(I) = C12$(3, N) THEN
A4$(I) = STR$(SA1(3, M, N))
ELSE
GOTO FGH7
END IF
FGH7:
NEXT N
NEXT I
HESAPT
SND2 = SND2 + 1
SONUC3(3, SND2) = SONUC2
ARAL(3, M, 2) = 0
GOTO DONGU2
ATANAN9:
ISARET = 0
SAYID = 0
SAYID3 = 0
ISARET11 = 0
UZ = LEN(UFG3$(3))
FOR NNN = 1 TO UZ
F1$ = MID$(UFG3$(3), NNN, 1)
IF F1$ = "(" THEN
ISARET11 = 0
SAYID3 = 0
LL = NNN + 1
LM = NNN

```

```

ISARET = ISARET + 1
A3$(ISARET) = F1$
ELSEIF F1$ = ")" THEN
ISARET = ISARET + 1
A3$(ISARET) = F1$
GOTO KARSİ8
ELSEIF F1$ = "*" OR F1$ = "/" OR F1$ =
"+" OR F1$ = "-" OR F1$ = "^" THEN
ISARET11 = ISARET11 + 1
ISARET = ISARET + 1
A3$(ISARET) = F1$
ELSE
SAYID3 = SAYID3 + 1
SAYID = SAYID + 1
A4$(SAYID) = F1$
END IF
DONE8:
NEXT NNN
GOTO SEAN8
KARSİ8:
IF ISARET11 <> 0 AND SAYID3 <> 0 AND
ISARET11 = SAYID3 THEN
EKSİİ
GOTO SANİ8
ELSE
GOTO DONE8
END IF
SANİ8:
FOR I = 1 TO SAYID3
IF A4$(I) = "X" THEN
A4$(I) = STR$(-X3)
ELSE
GOTO MLKN8
END IF
MLKN8:
NEXT I
GOTO DONE8
SEAN8:
IF SAYID <= 1 AND F1$ <> "X" THEN
SND2 = SND2 + 1
SONUC3(3, SND2) = SA2(3, M, 1)
ARAL(3, M, 3) = 0
GOTO DONGU2
ELSE
GOTO SAN8
END IF
SAN8:
FOR I = 1 TO SAYID
IF A4$(I) = "X" THEN
A4$(I) = STR$(X3)
ELSE
GOTO MLK8
END IF
MLK8:
NEXT I
FOR I = 1 TO SAYID

```

```

FOR N = 1 TO K13(3)
IF A4$(I) = C13$(3, N) THEN
A4$(I) = STR$(SA2(3, M, N))
ELSE
GOTO FGH8
END IF
FGH8:
NEXT N
NEXT I
HESAPT
SND2 = SND2 + 1
SONUC3(3, SND2) = SONUC2
ARAL(3, M, 3) = 0
DONGU2:
NEXT M
SAYI1 = 4
GOTO KONT
CASE 4
SND3 = 0
FOR M = 1 TO GUK(4)
IF ARAL(4, M, 1) = 1 THEN
GOTO ATANAN10
ELSE
GOTO SORAN7
END IF
SORAN7:
IF ARAL(4, M, 2) = 1 THEN
GOTO ATANAN11
ELSE
GOTO SORAN8
END IF
SORAN8:
IF ARAL(4, M, 3) = 1 THEN
GOTO ATANAN12
ELSE
GOTO DONGU3
END IF
ATANAN10:
ISARET = 0
SAYID = 0
SAYID3 = 0
ISARET11 = 0
UZ = LEN(UFG1$(4))
FOR NNN = 1 TO UZ
F1$ = MID$(UFG1$(4), NNN, 1)
IF F1$ = "(" THEN
ISARET11 = 0
SAYID3 = 0
LL = NNN + 1
LM = NNN
ISARET = ISARET + 1
A3$(ISARET) = F1$
ELSEIF F1$ = ")" THEN
ISARET = ISARET + 1
A3$(ISARET) = F1$
GOTO KARSİ9

```

```

ELSEIF F1$ = "*" OR F1$ = "/" OR F1$ =
"+" OR F1$ = "-" OR F1$ = "^" THEN
ISARET11 = ISARET11 + 1
ISARET = ISARET + 1
A3$(ISARET) = F1$
ELSE
SAYID3 = SAYID3 + 1
SAYID = SAYID + 1
A4$(SAYID) = F1$
END IF
DONE9:
NEXT NNN
GOTO SEAN9
KARSİ9:
IF ISARET11 <> 0 AND SAYID3 <> 0 AND
ISARET11 = SAYID3 THEN
EKSİİ
GOTO SANİ9
ELSE
GOTO DONE9
END IF
SANİ9:
FOR I = 1 TO SAYID3
IF A4$(I) = "X" THEN
A4$(I) = STR$(-X4)
ELSE
GOTO MLKN9
END IF
MLKN9:
NEXT I
GOTO DONE9
SEAN9:
IF SAYID <= 1 AND F1$ <> "X" THEN
SND3 = SND3 + 1
SONUC3(4, SND3) = SA(4, M, 1)
ARAL(4, M, 1) = 0
GOTO DONGU3
ELSE
GOTO SAN9
END IF
SAN9:
FOR I = 1 TO SAYID
IF A4$(I) = "X" THEN
A4$(I) = STR$(X4)
ELSEIF A4$(I) = STR$(-X4) THEN
GOTO MLK9
ELSE
GOTO MLK9
END IF
MLK9:
NEXT I
FOR I = 1 TO SAYID
FOR N = 1 TO K11(4)
IF A4$(I) = C11$(4, N) THEN
A4$(I) = STR$(SA(4, M, N))
ELSE

```

```

GOTO FGH9
END IF
FGH9:
NEXT N
NEXT I
HESAPT
SND3 = SND3 + 1
SONUC3(4, SND3) = SONUC2
ARAL(4, M, 1) = 0
GOTO DONGU3
ATANAN11:
ISARET = 0
SAYID = 0
SAYID3 = 0
ISARET11 = 0
UZ = LEN(UFG2$(4))
FOR NNN = 1 TO UZ
F1$ = MID$(UFG2$(4), NNN, 1)
IF F1$ = "(" THEN
ISARET11 = 0
SAYID3 = 0
LL = NNN + 1
LM = NNN
ISARET = ISARET + 1
A3$(ISARET) = F1$
ELSEIF F1$ = ")" THEN
ISARET = ISARET + 1
A3$(ISARET) = F1$
GOTO KARS10
ELSEIF F1$ = "*" OR F1$ = "/" OR F1$ =
"+" OR F1$ = "-" OR F1$ = "^" THEN
ISARET11 = ISARET11 + 1
ISARET = ISARET + 1
A3$(ISARET) = F1$
ELSE
SAYID3 = SAYID3 + 1
SAYID = SAYID + 1
A4$(SAYID) = F1$
END IF
DONE10:
NEXT NNN
GOTO SEAN10
KARS10:
IF ISARET11 <> 0 AND SAYID3 <> 0 AND
ISARET11 = SAYID3 THEN
EKSII
GOTO SAN10
ELSE
GOTO DONE10
END IF
SAN10:
FOR I = 1 TO SAYID3
IF A4$(I) = "X" THEN
A4$(I) = STR$(X4)
ELSE
GOTO MLKN10
END IF
MLKN10:
NEXT I
FOR I = 1 TO SAYID
FOR N = 1 TO K12(4)
IF A4$(I) = C12$(4, N) THEN
A4$(I) = STR$(SA1(4, M, N))
ELSE
GOTO FGH10
END IF
FGH10:
NEXT N
NEXT I
HESAPT
SND3 = SND3 + 1
SONUC3(4, SND3) = SONUC2
ARAL(4, M, 2) = 0
GOTO DONGU3
ATANAN12:
ISARET = 0
SAYID = 0
SAYID3 = 0
ISARET11 = 0
UZ = LEN(UFG3$(4))
FOR NNN = 1 TO UZ
F1$ = MID$(UFG3$(4), NNN, 1)
IF F1$ = "(" THEN
ISARET11 = 0
SAYID3 = 0
LL = NNN + 1
LM = NNN
ISARET = ISARET + 1
A3$(ISARET) = F1$
ELSEIF F1$ = ")" THEN
ISARET = ISARET + 1
A3$(ISARET) = F1$
GOTO KARS11

```

```

END IF
MLKN10:
NEXT I
GOTO DONE10
SEAN10:
IF SAYID <= 1 AND F1$ <> "X" THEN
SND3 = SND3 + 1
SONUC3(4, SND3) = SA1(4, M, 1)
ARAL(4, M, 2) = 0
GOTO DONGU3
ELSE
GOTO SAN10
END IF
SAN10:
FOR I = 1 TO SAYID
IF A4$(I) = "X" THEN
A4$(I) = STR$(X4)
ELSE
GOTO MLK10
END IF
MLK10:
NEXT I
FOR I = 1 TO SAYID
FOR N = 1 TO K12(4)
IF A4$(I) = C12$(4, N) THEN
A4$(I) = STR$(SA1(4, M, N))
ELSE
GOTO FGH10
END IF
FGH10:
NEXT N
NEXT I
HESAPT
SND3 = SND3 + 1
SONUC3(4, SND3) = SONUC2
ARAL(4, M, 2) = 0
GOTO DONGU3
ATANAN12:
ISARET = 0
SAYID = 0
SAYID3 = 0
ISARET11 = 0
UZ = LEN(UFG3$(4))
FOR NNN = 1 TO UZ
F1$ = MID$(UFG3$(4), NNN, 1)
IF F1$ = "(" THEN
ISARET11 = 0
SAYID3 = 0
LL = NNN + 1
LM = NNN
ISARET = ISARET + 1
A3$(ISARET) = F1$
ELSEIF F1$ = ")" THEN
ISARET = ISARET + 1
A3$(ISARET) = F1$
GOTO KARS11

```

```

ELSEIF F1$ = "*" OR F1$ = "/" OR F1$ =
"+" OR F1$ = "-" OR F1$ = "^" THEN
  ISARET11 = ISARET11 + 1
  ISARET = ISARET + 1
  A3$(ISARET) = F1$
ELSE
  SAYID3 = SAYID3 + 1
  SAYID = SAYID + 1
  A4$(SAYID) = F1$
END IF
DONE11:
NEXT NNN
GOTO SEAN11
KARS11:
IF ISARET11 <> 0 AND SAYID3 <> 0 AND
ISARET11 = SAYID3 THEN
  EKS11
GOTO SAN11
ELSE
  GOTO DONE11
END IF
SAN11:
FOR I = 1 TO SAYID3
  IF A4$(I) = "X" THEN
    A4$(I) = STR$(-X4)
  ELSE
    GOTO MLKN11
  END IF
MLKN11:
NEXT I
GOTO DONE11
SEAN11:
IF SAYID <= 1 AND F1$ <> "X" THEN
  SND3 = SND3 + 1
  SONUC3(4, SND3) = SA2(4, M, 1)
  ARAL(4, M, 3) = 0
  GOTO DONGU3
ELSE
  GOTO SAN11
END IF
SAN11:
FOR I = 1 TO SAYID
  IF A4$(I) = "X" THEN
    A4$(I) = STR$(X4)
  ELSE
    GOTO MLK11
  END IF
MLK11:
NEXT I
FOR I = 1 TO SAYID
  FOR N = 1 TO K13(4)
    IF A4$(I) = C13$(4, N) THEN
      A4$(I) = STR$(SA2(4, M, N))
    ELSE
      GOTO FGH11
    END IF

```

```

FGH11:
NEXT N
NEXT I
HESAPT
SND3 = SND3 + 1
SONUC3(4, SND3) = SONUC2
ARAL(4, M, 3) = 0
DONGU3:
NEXT M
SAYI1 = 5
GOTO KONT
CASE 5
  SND4 = 0
  FOR M = 1 TO GUK(5)
    IF ARAL(5, M, 1) = 1 THEN
      GOTO ATANAN13
    ELSE
      GOTO SORAN9
    END IF
  SORAN9:
    IF ARAL(5, M, 2) = 1 THEN
      GOTO ATANAN14
    ELSE
      GOTO SORAN10
    END IF
  SORAN10:
    IF ARAL(5, M, 3) = 1 THEN
      GOTO ATANAN15
    ELSE
      GOTO DONGU4
    END IF
  ATANAN13:
    ISARET = 0
    SAYID = 0
    SAYID3 = 0
    ISARET11 = 0
    UZ = LEN(UFG1$(5))
    FOR NNN = 1 TO UZ
      F1$ = MID$(UFG1$(5), NNN, 1)
      IF F1$ = "(" THEN
        ISARET11 = 0
        SAYID3 = 0
        LL = NNN + 1
        LM = NNN
        ISARET = ISARET + 1
        A3$(ISARET) = F1$
      ELSEIF F1$ = ")" THEN
        ISARET = ISARET + 1
        A3$(ISARET) = F1$
      GOTO KARS12
    ELSEIF F1$ = "*" OR F1$ = "/" OR F1$ =
      "+" OR F1$ = "-" OR F1$ = "^" THEN
      ISARET11 = ISARET11 + 1
      ISARET = ISARET + 1
      A3$(ISARET) = F1$
    ELSE

```

```

SAYID3 = SAYID3 + 1
SAYID = SAYID + 1
A4$(SAYID) = F1$
END IF
DONE12:
NEXT NNN
GOTO SEAN12
KARSİ12:
IF İSARET11 < 0 AND SAYID3 < 0 AND
İSARET11 = SAYID3 THEN
EKSİİ
GOTO SANİ12
ELSE
GOTO DONE12
END IF
SANİ12:
FOR İ = 1 TO SAYID3
IF A4$(İ) = "X" THEN
A4$(İ) = STR$(-X5)
ELSE
GOTO MLKN12
END IF
MLKN12:
NEXT İ
GOTO DONE12
SEAN12:
IF SAYID <= 1 AND F1$ < "X" THEN
SND4 = SND4 + 1
SONUC3(5, SND4) = SA(5, M, 1)
ARAL(5, M, 1) = 0
GOTO DONGU4
ELSE
GOTO SANİ12
END IF
SANİ12:
FOR İ = 1 TO SAYID
IF A4$(İ) = "X" THEN
A4$(İ) = STR$(X5)
ELSEIF A4$(İ) = STR$(-X5) THEN
GOTO MLK12
ELSE
GOTO MLK12
END IF
MLK12:
NEXT İ
FOR İ = 1 TO SAYID
FOR N = 1 TO K11(5)
IF A4$(İ) = C11$(5, N) THEN
A4$(İ) = STR$(SA(5, M, N))
ELSE
GOTO FGH12
END IF
FGH12:
NEXT N
NEXT İ
HESAPT

```

```

SND4 = SND4 + 1
SONUC3(5, SND4) = SONUC2
ARAL(5, M, 1) = 0
GOTO DONGU4
ATANAN14:
İSARET = 0
SAYID = 0
SAYID3 = 0
İSARET11 = 0
UZ = LEN(UFG2$(5))
FOR NNN = 1 TO UZ
F1$ = MID$(UFG2$(5), NNN, 1)
IF F1$ = "(" THEN
İSARET11 = 0
SAYID3 = 0
LL = NNN + 1
LM = NNN
İSARET = İSARET + 1
A3$(İSARET) = F1$
ELSEIF F1$ = ")" THEN
İSARET = İSARET + 1
A3$(İSARET) = F1$
GOTO KARSİ13
ELSEIF F1$ = "*" OR F1$ = "/" OR F1$ =
"+" OR F1$ = "-" OR F1$ = "^" THEN
İSARET11 = İSARET11 + 1
İSARET = İSARET + 1
A3$(İSARET) = F1$
ELSE
SAYID3 = SAYID3 + 1
SAYID = SAYID + 1
A4$(SAYID) = F1$
END IF
DONE13:
NEXT NNN
GOTO SEAN13
KARSİ13:
IF İSARET11 < 0 AND SAYID3 < 0 AND
İSARET11 = SAYID3 THEN
EKSİİ
GOTO SANİ13
ELSE
GOTO DONE13
END IF
SANİ13:
FOR İ = 1 TO SAYID3
IF A4$(İ) = "X" THEN
A4$(İ) = STR$(-X5)
ELSE
GOTO MLKN13
END IF
MLKN13:
NEXT İ
GOTO DONE13
SEAN13:
IF SAYID <= 1 AND F1$ < "X" THEN

```

```

SND4 = SND4 + 1
SONUC3(5, SND4) = SA1(5, M, 1)

ARAL(5, M, 2) = 0
GOTO DONGU4
ELSE
GOTO SAN13
END IF
SAN13:
FOR I = 1 TO SAYID
IF A4$(I) = "X" THEN
A4$(I) = STR$(X5)
ELSE
GOTO MLK13
END IF
MLK13:
NEXT I
FOR I = 1 TO SAYID
FOR N = 1 TO K12(5)
IF A4$(I) = C12$(5, N) THEN
A4$(I) = STR$(SA1(5, M, N))
ELSE
GOTO FGH13
END IF
FGH13:
NEXT N
NEXT I
HESAPT
SND4 = SND4 + 1
SONUC3(5, SND4) = SONUC2

ARAL(5, M, 2) = 0
GOTO DONGU4
ATANAN15:
ISARET = 0
SAYID = 0
SAYID3 = 0
ISARET11 = 0
UZ = LEN(UFG3$(5))
FOR NNN = 1 TO UZ
F1$ = MID$(UFG3$(5), NNN, 1)
IF F1$ = "(" THEN
ISARET11 = 0
SAYID3 = 0
LL = NNN + 1
LM = NNN
ISARET = ISARET + 1
A3$(ISARET) = F1$
ELSEIF F1$ = ")" THEN
ISARET = ISARET + 1
A3$(ISARET) = F1$
GOTO KARSİ14
ELSEIF F1$ = "*" OR F1$ = "/" OR F1$ =
"+" OR F1$ = "-" OR F1$ = "^" THEN
ISARET11 = ISARET11 + 1
ISARET = ISARET + 1

```

```

A3$(ISARET) = F1$
ELSE
SAYID3 = SAYID3 + 1
SAYID = SAYID + 1
A4$(SAYID) = F1$
END IF
DONE14:
NEXT NNN
GOTO SEAN14
KARSİ14:
IF ISARET11 < 0 AND SAYID3 < 0 AND
ISARET11 = SAYID3 THEN
EKSİİ
GOTO SANİ14
ELSE
GOTO DONE14
END IF
SANİ14:
FOR I = 1 TO SAYID3
IF A4$(I) = "X" THEN
A4$(I) = STR$(-X5)
ELSE
GOTO MLKN14
END IF
MLKN14:
NEXT I
GOTO DONE14
SEAN14:
IF SAYID <= 1 AND F1$ < "X" THEN
SND4 = SND4 + 1
SONUC3(5, SND4) = SA2(5, M, 1)
ARAL(5, M, 3) = 0
GOTO DONGU4
ELSE
GOTO SAN14
END IF
SAN14:
FOR I = 1 TO SAYID
IF A4$(I) = "X" THEN
A4$(I) = STR$(X5)
ELSE
GOTO MLK14
END IF
MLK14:
NEXT I
FOR I = 1 TO SAYID
FOR N = 1 TO K13(5)
IF A4$(I) = C13$(5, N) THEN
A4$(I) = STR$(SA2(5, M, N))
ELSE
GOTO FGH14
END IF
FGH14:
NEXT N
NEXT I
HESAPT

```

```

SND4 = SND4 + 1
SONUC3(5, SND4) = SONUC2
ARAL(5, M, 3) = 0
DONGU4:
NEXT M
SAYI1 = 6
GOTO KONT
REM *****ALTINCI GİRİŞ İÇİN
YAPILAN ATAMA-
LAR*****
CASE 6
SND5 = 0
FOR M = 1 TO GUK(6)
IF ARAL(6, M, 1) = 1 THEN
GOTO ATANAN16
ELSE
GOTO SORAN11
END IF
SORAN11:
IF ARAL(6, M, 2) = 1 THEN
GOTO ATANAN17
ELSE
GOTO SORAN12
END IF
SORAN12:
IF ARAL(2, M, 3) = 1 THEN
GOTO ATANAN18
ELSE
GOTO DONGU5
END IF
REM *****1. ARALIK İÇİN YAPILAN
ATAMALAR*****
ATANAN16:
İSARET = 0
SAYID = 0
SAYID3 = 0
İSARET11 = 0
UZ = LEN(UFG1$(6))
FOR NNN = 1 TO UZ
F1$ = MID$(UFG1$(6), NNN, 1)
IF F1$ = "(" THEN
İSARET11 = 0
SAYID3 = 0
LL = NNN + 1
LM = NNN
İSARET = İSARET + 1
A3$(İSARET) = F1$
ELSEIF F1$ = ")" THEN
İSARET = İSARET + 1
A3$(İSARET) = F1$
GOTO KARSİ15
ELSEIF F1$ = "*" OR F1$ = "/" OR F1$ =
"+" OR F1$ = "-" OR F1$ = "^" THEN
İSARET11 = İSARET11 + 1
İSARET = İSARET + 1
A3$(İSARET) = F1$

```

```

ELSE
SAYID3 = SAYID3 + 1
SAYID = SAYID + 1
A4$(SAYID) = F1$
END IF
DONE15:
NEXT NNN
GOTO SEAN15
KARSİ15:
IF İSARET11 <> 0 AND SAYID3 <> 0 AND
İSARET11 = SAYID3 THEN
EKSIİ
GOTO SANI15
ELSE
GOTO DONE15
END IF
SANI15:
FOR İ = 1 TO SAYID3
IF A4$(İ) = "X" THEN
A4$(İ) = STR$(-X6)
ELSE
GOTO MLKN15
END IF
MLKN15:
NEXT İ
GOTO DONE15
SEAN15:
IF SAYID <= 1 AND F1$ <> "X" THEN
SND5 = SND5 + 1
SONUC3(6, SND5) = SA(6, M, 1)
ARAL(6, M, 1) = 0
GOTO DONGU5
ELSE
GOTO SANI15
END IF
SANI15:
FOR İ = 1 TO SAYID
IF A4$(İ) = "X" THEN
A4$(İ) = STR$(X6)
ELSEIF A4$(İ) = STR$(-X6) THEN
GOTO MLK15
ELSE
GOTO MLK15
END IF
MLK15:
NEXT İ
FOR İ = 1 TO SAYID
FOR N = 1 TO K11(6)
IF A4$(İ) = C11$(6, N) THEN
A4$(İ) = STR$(SA(6, M, N))
ELSE
GOTO FGH15
END IF
FGH15:
NEXT N
NEXT İ

```



```

HESAPT
SND5 = SND5 + 1
SONUC3(6, SND5) = SONUC2
ARAL(6, M, 1) = 0
GOTO DONGU5
ATANAN17:
ISARET = 0
SAYID = 0
SAYID3 = 0
ISARET11 = 0
UZ = LEN(UFG2$(6))
FOR NNN = 1 TO UZ
F1$ = MID$(UFG2$(6), NNN, 1)
IF F1$ = "(" THEN
ISARET11 = 0
SAYID3 = 0
LL = NNN + 1
LM = NNN
ISARET = ISARET + 1
A3$(ISARET) = F1$
ELSEIF F1$ = ")" THEN
ISARET = ISARET + 1
A3$(ISARET) = F1$
GOTO KARSİ16
ELSEIF F1$ = "*" OR F1$ = "/" OR F1$ =
"+" OR F1$ = "-" OR F1$ = "^" THEN
ISARET11 = ISARET11 + 1
ISARET = ISARET + 1
A3$(ISARET) = F1$
ELSE
SAYID3 = SAYID3 + 1
SAYID = SAYID + 1
A4$(SAYID) = F1$
END IF
DONE16:
NEXT NNN
GOTO SEAN16
KARSİ16:
IF ISARET11 <> 0 AND SAYID3 <> 0 AND
ISARET11 = SAYID3 THEN
EKSİİ
GOTO SANİ16
ELSE
GOTO DONE16
END IF
SANİ16:
FOR İ = 1 TO SAYID3
IF A4$(İ) = "X" THEN
A4$(İ) = STR$(X6)
ELSE
GOTO MLKN16
END IF
MLKN16:
NEXT İ
GOTO DONE16
SEAN16:

```

```

IF SAYID <= 1 AND F1$ <> "X" THEN
SND5 = SND5 + 1
SONUC3(6, SND5) = SA1(6, M, 1)
ARAL(6, M, 2) = 0
GOTO DONGU5
ELSE
GOTO SAN16
END IF
SAN16:
FOR İ = 1 TO SAYID
IF A4$(İ) = "X" THEN
A4$(İ) = STR$(X6)
ELSE
GOTO MLK16
END IF
MLK16:
NEXT İ
FOR İ = 1 TO SAYID
FOR N = 1 TO K12(6)
IF A4$(İ) = C12$(6, N) THEN
A4$(İ) = STR$(SA1(6, M, N))
ELSE
GOTO FGH16
END IF
FGH16:
NEXT N
NEXT İ
HESAPT
SND5 = SND5 + 1
SONUC3(6, SND5) = SONUC2

ARAL(6, M, 2) = 0
GOTO DONGU5
ATANAN18:
ISARET = 0
SAYID = 0
SAYID3 = 0
ISARET11 = 0
UZ = LEN(UFG3$(6))
FOR NNN = 1 TO UZ
F1$ = MID$(UFG3$(6), NNN, 1)
IF F1$ = "(" THEN
ISARET11 = 0
SAYID3 = 0
LL = NNN + 1
LM = NNN
ISARET = ISARET + 1
A3$(ISARET) = F1$
ELSEIF F1$ = ")" THEN
ISARET = ISARET + 1
A3$(ISARET) = F1$
GOTO KARSİ17
ELSEIF F1$ = "*" OR F1$ = "/" OR F1$ =
"+" OR F1$ = "-" OR F1$ = "^" THEN
ISARET11 = ISARET11 + 1
ISARET = ISARET + 1

```

EK-1

```

A3$(ISARET) = F1$
ELSE
SAYID3 = SAYID3 + 1
SAYID = SAYID + 1
A4$(SAYID) = F1$
END IF
DONE17:
NEXT NNN
GOTO SEAN17
KARS17:
IF ISARET11 < 0 AND SAYID3 < 0 AND
ISARET11 = SAYID3 THEN
EKS11
GOTO SAN17
ELSE
GOTO DONE17
END IF
SAN17:
FOR I = 1 TO SAYID3
IF A4$(I) = "X" THEN
A4$(I) = STR$(-X6)
ELSE
GOTO MLKN17
END IF
MLKN17:
NEXT I
GOTO DONE17
SEAN17:
IF SAYID <= 1 AND F1$ < "X" THEN
SND5 = SND5 + 1
SONUC3(6, SND5) = SA2(6, M, 1)
ARAL(6, M, 3) = 0
GOTO DONGU5
ELSE
GOTO SAN17
END IF
SAN17:
FOR I = 1 TO SAYID
IF A4$(I) = "X" THEN
A4$(I) = STR$(X6)
ELSE
GOTO MLK17
END IF
MLK17:
NEXT I
FOR I = 1 TO SAYID
FOR N = 1 TO K13(6)
IF A4$(I) = C13$(6, N) THEN
A4$(I) = STR$(SA2(6, M, N))
ELSE
GOTO FGH17
END IF
FGH17:
NEXT N
NEXT I
HESAPT

```

```

SND5 = SND5 + 1
SONUC3(6, SND5) = SONUC2
ARAL(6, M, 3) = 0
DONGU5:
NEXT M
SAYI1 = 7
GOTO KONT
CASE 7
SND6 = 0
FOR M = 1 TO GUK(7)
IF ARAL(7, M, 1) = 1 THEN
GOTO ATANAN19
ELSE
GOTO SORAN13
END IF
SORAN13:
IF ARAL(7, M, 2) = 1 THEN
GOTO ATANAN20
ELSE
GOTO SORAN14
END IF
SORAN14:
IF ARAL(7, M, 3) = 1 THEN
GOTO ATANAN21
ELSE
GOTO DONGU6
END IF
ATANAN19:
ISARET = 0
SAYID = 0
SAYID3 = 0
ISARET11 = 0
UZ = LEN(UFG1$(7))
FOR NNN = 1 TO UZ
F1$ = MID$(UFG1$(7), NNN, 1)
IF F1$ = "(" THEN
ISARET11 = 0
SAYID3 = 0
LL = NNN + 1
LM = NNN
ISARET = ISARET + 1
A3$(ISARET) = F1$
ELSEIF F1$ = ")" THEN
ISARET = ISARET + 1
A3$(ISARET) = F1$
GOTO KARS18
ELSEIF F1$ = "*" OR F1$ = "/" OR F1$ =
"+" OR F1$ = "-" OR F1$ = "^" THEN
ISARET11 = ISARET11 + 1
ISARET = ISARET + 1
A3$(ISARET) = F1$
ELSE
SAYID3 = SAYID3 + 1
SAYID = SAYID + 1
A4$(SAYID) = F1$
END IF

```

```

DONE18:
NEXT NNN
GOTO SEAN18
KARSİ18:
IF İSARET11 < 0 AND SAYİD3 < 0 AND
İSARET11 = SAYİD3 THEN
EKSİİ
GOTO SANİ18
ELSE
GOTO DONE18
END IF
SANİ18:
FOR İ = 1 TO SAYİD3
IF A4$(İ) = "X" THEN
A4$(İ) = STR$(-X7)
ELSE
GOTO MLKN18
END IF
MLKN18:
NEXT İ
GOTO DONE18
SEAN18:
IF SAYİD <= 1 AND F1$ < "X" THEN
SND6 = SND6 + 1
SONUC3(7, SND6) = SA(7, M, 1)
ARAL(7, M, 1) = 0
GOTO DONGU6
ELSE
GOTO SANİ18
END IF
SANİ18:
FOR İ = 1 TO SAYİD
IF A4$(İ) = "X" THEN
A4$(İ) = STR$(X7)
ELSEIF A4$(İ) = STR$(-X7) THEN ,
GOTO MLK18
ELSE
GOTO MLK18
END IF
MLK18:
NEXT İ
FOR İ = 1 TO SAYİD
FOR N = 1 TO K11(7)
IF A4$(İ) = C11$(7, N) THEN
A4$(İ) = STR$(SA(7, M, N))
ELSE
GOTO FGH18
END IF
FGH18:
NEXT N
NEXT İ
HESAPT
SND6 = SND6 + 1
SONUC3(7, SND6) = SONUC2
ARAL(7, M, 1) = 0
GOTO DONGU6

```

```

ATANAN20:
İSARET = 0
SAYİD = 0
SAYİD3 = 0
İSARET11 = 0
UZ = LEN(UFG2$(7))
FOR NNN = 1 TO UZ
F1$ = MID$(UFG2$(7), NNN, 1)
IF F1$ = "(" THEN
İSARET11 = 0
SAYİD3 = 0
LL = NNN + 1
LM = NNN
İSARET = İSARET + 1
A3$(İSARET) = F1$
ELSEIF F1$ = ")" THEN
İSARET = İSARET + 1
A3$(İSARET) = F1$
GOTO KARSİ19
ELSEIF F1$ = "*" OR F1$ = "/" OR F1$ =
"+" OR F1$ = "-" OR F1$ = "^" THEN
İSARET11 = İSARET11 + 1
İSARET = İSARET + 1
A3$(İSARET) = F1$
ELSE
SAYİD3 = SAYİD3 + 1
SAYİD = SAYİD + 1
A4$(SAYİD) = F1$
END IF
DONE19:
NEXT NNN
GOTO SEAN19
KARSİ19:
IF İSARET11 < 0 AND SAYİD3 < 0 AND
İSARET11 = SAYİD3 THEN
EKSİİ
GOTO SANİ19
ELSE
GOTO DONE19
END IF
SANİ19:
FOR İ = 1 TO SAYİD3
IF A4$(İ) = "X" THEN
A4$(İ) = STR$(-X7)
ELSE
GOTO MLKN19
END IF
MLKN19:
NEXT İ
GOTO DONE19
SEAN19:
IF SAYİD <= 1 AND F1$ < "X" THEN
SND6 = SND6 + 1
SONUC3(7, SND6) = SA1(7, M, 1)
ARAL(7, M, 2) = 0
GOTO DONGU6

```

```

ELSE
GOTO SAN19
END IF
SAN19:
FOR I = 1 TO SAYID
IF A4$(I) = "X" THEN
A4$(I) = STR$(X7)
ELSE
GOTO MLK19
END IF
MLK19:
NEXT I
FOR I = 1 TO SAYID
FOR N = 1 TO K12(7)
IF A4$(I) = C12$(7, N) THEN
A4$(I) = STR$(SA1(7, M, N))
ELSE
GOTO FGH19
END IF
FGH19:
NEXT N
NEXT I
HESAPT
SND6 = SND6 + 1
SONUC3(7, SND6) = SONUC2
ARAL(7, M, 2) = 0
GOTO DONGU6
ATANAN21:
ISARET = 0
SAYID = 0
SAYID3 = 0
ISARET11 = 0
UZ = LEN(UFG3$(7))
FOR NNN = 1 TO UZ
F1$ = MID$(UFG3$(7), NNN, 1)
IF F1$ = "(" THEN
ISARET11 = 0
SAYID3 = 0
LL = NNN + 1
LM = NNN
ISARET = ISARET + 1
A3$(ISARET) = F1$
ELSEIF F1$ = ")" THEN
ISARET = ISARET + 1
A3$(ISARET) = F1$
GOTO KARSİ20
ELSEIF F1$ = "*" OR F1$ = "/" OR F1$ =
"+" OR F1$ = "-" OR F1$ = "^" THEN
ISARET11 = ISARET11 + 1
ISARET = ISARET + 1
A3$(ISARET) = F1$
ELSE
SAYID3 = SAYID3 + 1
SAYID = SAYID + 1
A4$(SAYID) = F1$
END IF

```

```

DONE20:
NEXT NNN
GOTO SEAN20
KARSİ20:
IF ISARET11 <> 0 AND SAYID3 <> 0 AND
ISARET11 = SAYID3 THEN
EKSiİ
GOTO SANİ20
ELSE
GOTO DONE20
END IF
SANİ20:
FOR I = 1 TO SAYID3
IF A4$(I) = "X" THEN
A4$(I) = STR$(-X7)
ELSE
GOTO MLKN20
END IF
MLKN20:
NEXT I
GOTO DONE20
SEAN20:
IF SAYID <= 1 AND F1$ <> "X" THEN
SND6 = SND6 + 1
SONUC3(7, SND6) = SA2(7, M, 1)

ARAL(7, M, 3) = 0
GOTO DONGU6
ELSE
GOTO SAN20
END IF
SAN20:
FOR I = 1 TO SAYID
IF A4$(I) = "X" THEN
A4$(I) = STR$(X7)
ELSE
GOTO MLK20
END IF
MLK20:
NEXT I
FOR I = 1 TO SAYID
FOR N = 1 TO K13(7)
IF A4$(I) = C13$(7, N) THEN
A4$(I) = STR$(SA2(7, M, N))
ELSE
GOTO FGH20
END IF
FGH20:
NEXT N
NEXT I
HESAPT
SND6 = SND6 + 1
SONUC3(7, SND6) = SONUC2
ARAL(7, M, 3) = 0
DONGU6:
NEXT M

```

```

SAYİ1 = 8
GOTO KONT
CASE 8
SND7 = 0
FOR M = 1 TO GUK(8)
IF ARAL(8, M, 1) = 1 THEN
GOTO ATANAN22
ELSE
GOTO SORAN15
END IF
SORAN15:
IF ARAL(8, M, 2) = 1 THEN
GOTO ATANAN23
ELSE
GOTO SORAN16
END IF
SORAN16:
IF ARAL(8, M, 3) = 1 THEN
GOTO ATANAN24
ELSE
GOTO DONGU7
END IF
ATANAN22:
İSARET = 0
SAYİD = 0
SAYİD3 = 0
İSARET11 = 0
UZ = LEN(UFG1$(8))
FOR NNN = 1 TO UZ
F1$ = MID$(UFG1$(8), NNN, 1)
IF F1$ = "(" THEN
İSARET11 = 0
SAYİD3 = 0
LL = NNN + 1
LM = NNN
İSARET = İSARET + 1
A3$(İSARET) = F1$
ELSEIF F1$ = ")" THEN
İSARET = İSARET + 1
A3$(İSARET) = F1$
GOTO KARSİ21
ELSEIF F1$ = "*" OR F1$ = "/" OR F1$ =
"+" OR F1$ = "-" OR F1$ = "^" THEN
İSARET11 = İSARET11 + 1
İSARET = İSARET + 1
A3$(İSARET) = F1$
ELSE
SAYİD3 = SAYİD3 + 1
SAYİD = SAYİD + 1
A4$(SAYİD) = F1$
END IF
DONE21:
NEXT NNN
GOTO SEAN21
KARSİ21:

```

```

IF İSARET11 <> 0 AND SAYİD3 <> 0 AND
İSARET11 = SAYİD3 THEN
EKSİİ
GOTO SANİ21
ELSE
GOTO DONE21
END IF
SANİ21:
FOR İ = 1 TO SAYİD3
IF A4$(İ) = "X" THEN
A4$(İ) = STR$(-X8)
ELSE
GOTO MLKN21
END IF
MLKN21:
NEXT İ
GOTO DONE21
SEAN21:
IF SAYİD <= 1 AND F1$ <> "X" THEN
SND7 = SND7 + 1
SONUC3(8, SND7) = SA(8, M, 1)
ARAL(8, M, 1) = 0
GOTO DONGU7
ELSE
GOTO SAN21
END IF
SAN21:
FOR İ = 1 TO SAYİD
IF A4$(İ) = "X" THEN
A4$(İ) = STR$(X8)
ELSEIF A4$(İ) = STR$(-X8) THEN
GOTO MLK21
ELSE
GOTO MLK21
END IF
MLK21:
NEXT İ
FOR İ = 1 TO SAYİD
FOR N = 1 TO K11(8)
IF A4$(İ) = C11$(8, N) THEN
A4$(İ) = STR$(SA(8, M, N))
ELSE
GOTO FGH21
END IF
FGH21:
NEXT N
NEXT İ
HESAPT
SND7 = SND7 + 1
SONUC3(8, SND7) = SONUC2

ARAL(8, M, 1) = 0
GOTO DONGU7
ATANAN23:
İSARET = 0
SAYİD = 0

```

```

SAYID3 = 0
ISARET11 = 0
UZ = LEN(UFG2$(8))
FOR NNN = 1 TO UZ
F1$ = MID$(UFG2$(8), NNN, 1)
IF F1$ = "(" THEN
ISARET11 = 0
SAYID3 = 0
LL = NNN + 1
LM = NNN
ISARET = ISARET + 1
A3$(ISARET) = F1$
ELSEIF F1$ = ")" THEN
ISARET = ISARET + 1
A3$(ISARET) = F1$
GOTO KARS122
ELSEIF F1$ = "*" OR F1$ = "/" OR F1$ =
"+" OR F1$ = "-" OR F1$ = "^" THEN
ISARET11 = ISARET11 + 1
ISARET = ISARET + 1
A3$(ISARET) = F1$
ELSE
SAYID3 = SAYID3 + 1
SAYID = SAYID + 1
A4$(SAYID) = F1$
END IF
DONE22:
NEXT NNN
GOTO SEAN22
KARS122:
IF ISARET11 <> 0 AND SAYID3 <> 0 AND
ISARET11 = SAYID3 THEN
EKS11
GOTO SAN122
ELSE
GOTO DONE22
END IF
SAN122:
FOR I = 1 TO SAYID3
IF A4$(I) = "X" THEN
A4$(I) = STR$(-X8)
ELSE
GOTO MLKN22
END IF
MLKN22:
NEXT I
GOTO DONE22
SEAN22:
IF SAYID <= 1 AND F1$ <> "X" THEN
SND7 = SND7 + 1
SONUC3(8, SND7) = SA1(8, M, 1)
ARAL(8, M, 2) = 0
GOTO DONGU7
ELSE
GOTO SAN22
END IF

```

```

SAN22:
FOR I = 1 TO SAYID
IF A4$(I) = "X" THEN
A4$(I) = STR$(X8)
ELSE
GOTO MLK22
END IF
MLK22:
NEXT I
FOR I = 1 TO SAYID
FOR N = 1 TO K12(8)
IF A4$(I) = C12$(8, N) THEN
A4$(I) = STR$(SA1(8, M, N))
ELSE
GOTO FGH22
END IF
FGH22:
NEXT N
NEXT I
HESAPT
SND7 = SND7 + 1
SONUC3(8, SND7) = SONUC2
ARAL(8, M, 2) = 0
GOTO DONGU7
ATANAN24:
ISARET = 0
SAYID = 0
SAYID3 = 0
ISARET11 = 0
UZ = LEN(UFG3$(8))
FOR NNN = 1 TO UZ
F1$ = MID$(UFG3$(8), NNN, 1)
IF F1$ = "(" THEN
ISARET11 = 0
SAYID3 = 0
LL = NNN + 1
LM = NNN
ISARET = ISARET + 1
A3$(ISARET) = F1$
ELSEIF F1$ = ")" THEN
ISARET = ISARET + 1
A3$(ISARET) = F1$
GOTO KARS123
ELSEIF F1$ = "*" OR F1$ = "/" OR F1$ =
"+" OR F1$ = "-" OR F1$ = "^" THEN
ISARET11 = ISARET11 + 1
ISARET = ISARET + 1
A3$(ISARET) = F1$
ELSE
SAYID3 = SAYID3 + 1
SAYID = SAYID + 1
A4$(SAYID) = F1$
END IF
DONE23:
NEXT NNN
GOTO SEAN23

```

```

KARSİ23:
IF İSARET11 <> 0 AND SAYİD3 <> 0 AND
İSARET11 = SAYİD3 THEN
EKSİİ
GOTO SANİ23
ELSE
GOTO DONE23
END IF
SANİ23:
FOR İ = 1 TO SAYİD3
IF A4$(İ) = "X" THEN
A4$(İ) = STR$(-X8)
ELSE
GOTO MLKN23
END IF
MLKN23:
NEXT İ
GOTO DONE23
SEAN23:
IF SAYİD <= 1 AND F1$ <> "X" THEN
SND7 = SND7 + 1
SONUC3(8, SND7) = SA2(8, M, 1)
ARAL(8, M, 3) = 0
GOTO DONGU7
ELSE
GOTO SAN23
END IF
SAN23:
FOR İ = 1 TO SAYİD
IF A4$(İ) = "X" THEN
A4$(İ) = STR$(X8)
ELSE
GOTO MLK23
END IF
MLK23:
NEXT İ
FOR İ = 1 TO SAYİD
FOR N = 1 TO K13(8)
IF A4$(İ) = C13$(8, N) THEN
A4$(İ) = STR$(SA2(8, M, N))
ELSE
GOTO FGH23
END IF
FGH23:
NEXT N
NEXT İ
HESAPT
SND7 = SND7 + 1
SONUC3(8, SND7) = SONUC2
ARAL(8, M, 3) = 0
DONGU7:
NEXT M
SAYİ1 = 9
GOTO KONT
CASE 9

```

```

SND8 = 0
FOR M = 1 TO GUK(9)
IF ARAL(9, M, 1) = 1 THEN
GOTO ATANAN25
ELSE
GOTO SORAN17
END IF
SORAN17:
IF ARAL(9, M, 2) = 1 THEN
GOTO ATANAN26
ELSE
GOTO SORAN18
END IF
SORAN18:
IF ARAL(9, M, 3) = 1 THEN
GOTO ATANAN27
ELSE
GOTO DONGU8
END IF
ATANAN25:
İSARET = 0
SAYİD = 0
SAYİD3 = 0
İSARET11 = 0
UZ = LEN(UFG1$(9))
FOR NNN = 1 TO UZ
F1$ = MID$(UFG1$(9), NNN, 1)
IF F1$ = "(" THEN
İSARET11 = 0
SAYİD3 = 0
LL = NNN + 1
LM = NNN
İSARET = İSARET + 1
A3$(İSARET) = F1$
ELSEIF F1$ = ")" THEN
İSARET = İSARET + 1
A3$(İSARET) = F1$
GOTO KARSİ24
ELSEIF F1$ = "*" OR F1$ = "/" OR F1$ =
"+" OR F1$ = "-" OR F1$ = "^" THEN
İSARET11 = İSARET11 + 1
İSARET = İSARET + 1
A3$(İSARET) = F1$
ELSE
SAYİD3 = SAYİD3 + 1
SAYİD = SAYİD + 1
A4$(SAYİD) = F1$
END IF
DONE24:
NEXT NNN
GOTO SEAN24
KARSİ24:
IF İSARET11 <> 0 AND SAYİD3 <> 0 AND
İSARET11 = SAYİD3 THEN
EKSİİ
GOTO SANİ24

```

```

ELSE
GOTO DONE24
END IF
SANI24:
FOR I = 1 TO SAYID3
IF A4$(I) = "X" THEN
A4$(I) = STR$(-X9)
ELSE
GOTO MLKN24
END IF
MLKN24:
NEXT I
GOTO DONE24
SEAN24:
IF SAYID <= 1 AND F1$ <> "X" THEN
SND8 = SND8 + 1
SONUC3(9, SND8) = SA(9, M, 1)
ARAL(9, M, 1) = 0
GOTO DONGU8
ELSE
GOTO SAN24
END IF
SAN24:
FOR I = 1 TO SAYID
IF A4$(I) = "X" THEN
A4$(I) = STR$(X9)
ELSEIF A4$(I) = STR$(-X9) THEN
GOTO MLK24
ELSE
GOTO MLK24
END IF
MLK24:
NEXT I
FOR I = 1 TO SAYID
FOR N = 1 TO K11(9)
IF A4$(I) = C11$(9, N) THEN
A4$(I) = STR$(SA(9, M, N))
ELSE
GOTO FGH24
END IF
FGH24:
NEXT N
NEXT I
HESAPT
SND8 = SND8 + 1
SONUC3(9, SND8) = SONUC2
ARAL(9, M, 1) = 0
GOTO DONGU8
ATANAN26:
ISARET = 0
SAYID = 0
SAYID3 = 0
ISARET11 = 0
UZ = LEN(UFG2$(9))
FOR NNN = 1 TO UZ
F1$ = MID$(UFG2$(9), NNN, 1)

```

```

IF F1$ = "(" THEN
ISARET11 = 0
SAYID3 = 0
LL = NNN + 1
LM = NNN
ISARET = ISARET + 1
A3$(ISARET) = F1$
ELSEIF F1$ = ")" THEN
ISARET = ISARET + 1
A3$(ISARET) = F1$
GOTO KARSİ25
ELSEIF F1$ = "*" OR F1$ = "/" OR F1$ =
"+" OR F1$ = "-" OR F1$ = "^" THEN
ISARET11 = ISARET11 + 1
ISARET = ISARET + 1
A3$(ISARET) = F1$
ELSE
SAYID3 = SAYID3 + 1
SAYID = SAYID + 1
A4$(SAYID) = F1$
END IF
DONE25:
NEXT NNN
GOTO SEAN25
KARSİ25:
IF ISARET11 < 0 AND SAYID3 < 0 AND
ISARET11 = SAYID3 THEN
EKSİİ
GOTO SANİ25
ELSE
GOTO DONE25
END IF
SANİ25:
FOR I = 1 TO SAYID3
IF A4$(I) = "X" THEN
A4$(I) = STR$(-X9)
ELSE
GOTO MLKN25
END IF
MLKN25:
NEXT I
GOTO DONE25
SEAN25:
IF SAYID <= 1 AND F1$ <> "X" THEN
SND8 = SND8 + 1
SONUC3(9, SND8) = SA1(9, M, 1)
ARAL(9, M, 2) = 0
GOTO DONGU8
ELSE
GOTO SAN25
END IF
SAN25:
FOR I = 1 TO SAYID
IF A4$(I) = "X" THEN
A4$(I) = STR$(X9)
ELSE

```



```

GOTO MLK25
END IF
MLK25:
NEXT I
FOR I = 1 TO SAYID
FOR N = 1 TO K12(9)
IF A4$(I) = C12$(9, N) THEN
A4$(I) = STR$(SA1(9, M, N))
ELSE
GOTO FGH25
END IF
FGH25:
NEXT N
NEXT I
HESAPT
SND8 = SND8 + 1
SONUC3(9, SND8) = SONUC2
ARAL(9, M, 2) = 0
GOTO DONGU8
ATANAN27:
ISARET = 0
SAYID = 0
SAYID3 = 0
ISARET11 = 0
UZ = LEN(UFG3$(9))
FOR NNN = 1 TO UZ
F1$ = MID$(UFG3$(9), NNN, 1)
IF F1$ = "(" THEN
ISARET11 = 0
SAYID3 = 0
LL = NNN + 1
LM = NNN
ISARET = ISARET + 1
A3$(ISARET) = F1$
ELSEIF F1$ = ")" THEN
ISARET = ISARET + 1
A3$(ISARET) = F1$
GOTO KARSİ26
ELSEIF F1$ = "*" OR F1$ = "/" OR F1$ =
"+" OR F1$ = "-" OR F1$ = "^" THEN
ISARET11 = ISARET11 + 1
ISARET = ISARET + 1
A3$(ISARET) = F1$
ELSE
SAYID3 = SAYID3 + 1
SAYID = SAYID + 1
A4$(SAYID) = F1$
END IF
DONE26:
NEXT NNN
GOTO SEAN26
KARSİ26:
IF ISARET11 <> 0 AND SAYID3 <> 0 AND
ISARET11 = SAYID3 THEN
EKSİİ
GOTO SANİ26

```

```

ELSE
GOTO DONE26
END IF
SANİ26:
FOR I = 1 TO SAYID3
IF A4$(I) = "X" THEN
A4$(I) = STR$(-X9)
ELSE
GOTO MLKN26
END IF
MLKN26:
NEXT I
GOTO DONE26
SEAN26:
IF SAYID <= 1 AND F1$ <> "X" THEN
SND8 = SND8 + 1
SONUC3(9, SND8) = SA2(9, M, 1)
ARAL(9, M, 3) = 0
GOTO DONGU8
ELSE
GOTO SAN26
END IF
SAN26:
FOR I = 1 TO SAYID
IF A4$(I) = "X" THEN
A4$(I) = STR$(X9)
ELSE
GOTO MLK26
END IF
MLK26:
NEXT I
FOR I = 1 TO SAYID
FOR N = 1 TO K13(9)
IF A4$(I) = C13$(9, N) THEN
A4$(I) = STR$(SA2(9, M, N))
ELSE
GOTO FGH26
END IF
FGH26:
NEXT N
NEXT I
HESAPT
SND8 = SND8 + 1
SONUC3(9, SND8) = SONUC2
ARAL(9, M, 3) = 0
DONGU8:
NEXT M
SAYİİ = 10
GOTO KONT
CASE 10
SND9 = 0
FOR M = 1 TO GUK(10)
IF ARAL(10, M, 1) = 1 THEN
GOTO ATANAN28
ELSE
GOTO SORAN19

```

```

END IF
SORAN19:
IF ARAL(10, M, 2) = 1 THEN
GOTO ATANAN29
ELSE
GOTO SORAN20
END IF
SORAN20:
IF ARAL(10, M, 3) = 1 THEN
GOTO ATANAN30
ELSE
GOTO DONGU9
END IF
ATANAN28:
ISARET = 0
SAYID = 0
SAYID3 = 0
ISARET11 = 0
UZ = LEN(UFG1$(10))
FOR NNN = 1 TO UZ
F1$ = MID$(UFG1$(10), NNN, 1)
IF F1$ = "(" THEN
ISARET11 = 0
SAYID3 = 0
LL = NNN + 1
LM = NNN
ISARET = ISARET + 1
A3$(ISARET) = F1$
ELSEIF F1$ = ")" THEN
ISARET = ISARET + 1
A3$(ISARET) = F1$
GOTO KARSİ27
ELSEIF F1$ = "*" OR F1$ = "/" OR F1$ =
"+" OR F1$ = "-" OR F1$ = "^" THEN
ISARET11 = ISARET11 + 1
ISARET = ISARET + 1
A3$(ISARET) = F1$
ELSE
SAYID3 = SAYID3 + 1
SAYID = SAYID + 1
A4$(SAYID) = F1$
END IF
DONE27:
NEXT NNN
GOTO SEAN27
KARSİ27:
IF ISARET11 < 0 AND SAYID3 < 0 AND
ISARET11 = SAYID3 THEN
EKSİİ
GOTO SANİ27
ELSE
GOTO DONE27
END IF
SANİ27:
FOR İ = 1 TO SAYID3
IF A4$(İ) = "X" THEN

```

```

A4$(İ) = STR$(-X10)
ELSE
GOTO MLKN27
END IF
MLKN27:
NEXT İ
GOTO DONE27
SEAN27:
IF SAYID <= 1 AND F1$ <> "X" THEN
SND9 = SND9 + 1
SONUC3(10, SND9) = SA(10, M, 1)
ARAL(10, M, 1) = 0
GOTO DONGU9
ELSE
GOTO SAN27
END IF
SAN27:
FOR İ = 1 TO SAYID
IF A4$(İ) = "X" THEN
A4$(İ) = STR$(X10)
ELSEIF A4$(İ) = STR$(-X10) THEN
GOTO MLK27
ELSE
GOTO MLK27
END IF
MLK27:
NEXT İ
FOR İ = 1 TO SAYID
FOR N = 1 TO K11(10)
IF A4$(İ) = C11$(10, N) THEN
A4$(İ) = STR$(SA(10, M, N))
ELSE
GOTO FGH27
END IF
FGH27:
NEXT N
NEXT İ
HESAPT
SND9 = SND9 + 1
SONUC3(10, SND9) = SONUC2
ARAL(10, M, 1) = 0
GOTO DONGU9
ATANAN29:
ISARET = 0
SAYID = 0
SAYID3 = 0
ISARET11 = 0
UZ = LEN(UFG2$(10))
FOR NNN = 1 TO UZ
F1$ = MID$(UFG2$(10), NNN, 1)
IF F1$ = "(" THEN
ISARET11 = 0
SAYID3 = 0
LL = NNN + 1
LM = NNN
ISARET = ISARET + 1

```

```

A3$(ISARET) = F1$
ELSEIF F1$ = ")" THEN
ISARET = ISARET + 1
A3$(ISARET) = F1$
GOTO KARSİ28
ELSEIF F1$ = "*" OR F1$ = "/" OR F1$ =
"+" OR F1$ = "-" OR F1$ = "^" THEN
ISARET11 = ISARET11 + 1
ISARET = ISARET + 1
A3$(ISARET) = F1$
ELSE
SAYİD3 = SAYİD3 + 1
SAYİD = SAYİD + 1
A4$(SAYİD) = F1$
END IF
DONE28:
NEXT NNN
GOTO SEAN28
KARSİ28:
IF ISARET11 < 0 AND SAYİD3 < 0 AND
ISARET11 = SAYİD3 THEN
EKSİİ
GOTO SANİ28
ELSE
GOTO DONE28
END IF
SANİ28:
FOR İ = 1 TO SAYİD3
IF A4$(İ) = "X" THEN
A4$(İ) = STR$(X10)
ELSE
GOTO MLKN28
END IF
MLKN28:
NEXT İ
GOTO DONE28
SEAN28:
IF SAYİD <= 1 AND F1$ < "X" THEN
SND9 = SND9 + 1
SONUC3(10, SND9) = SA1(10, M, 1)
ARAL(10, M, 2) = 0
GOTO DONGU9
ELSE
GOTO SAN28
END IF
SAN28:
FOR İ = 1 TO SAYİD
IF A4$(İ) = "X" THEN
A4$(İ) = STR$(X10)
ELSE
GOTO MLK28
END IF
MLK28:
NEXT İ
FOR İ = 1 TO SAYİD
FOR N = 1 TO K12(10)

```

```

IF A4$(İ) = C12$(10, N) THEN
A4$(İ) = STR$(SA1(10, M, N))
ELSE
GOTO FGH28
END IF
FGH28:
NEXT N
NEXT İ
HESAPT
SND9 = SND9 + 1
SONUC3(10, SND9) = SONUC2
ARAL(10, M, 2) = 0
GOTO DONGU9
ATANAN30:
ISARET = 0
SAYİD = 0
SAYİD3 = 0
ISARET11 = 0
UZ = LEN(UFG3$(10))
FOR NNN = 1 TO UZ
F1$ = MID$(UFG3$(10), NNN, 1)
IF F1$ = "(" THEN
ISARET11 = 0
SAYİD3 = 0
LL = NNN + 1
LM = NNN
ISARET = ISARET + 1
A3$(ISARET) = F1$
ELSEIF F1$ = ")" THEN
ISARET = ISARET + 1
A3$(ISARET) = F1$
GOTO KARSİ29
ELSEIF F1$ = "*" OR F1$ = "/" OR F1$ =
"+" OR F1$ = "-" OR F1$ = "^" THEN
ISARET11 = ISARET11 + 1
ISARET = ISARET + 1
A3$(ISARET) = F1$
ELSE
SAYİD3 = SAYİD3 + 1
SAYİD = SAYİD + 1
A4$(SAYİD) = F1$
END IF
DONE29:
NEXT NNN
GOTO SEAN29
KARSİ29:
IF ISARET11 < 0 AND SAYİD3 < 0 AND
ISARET11 = SAYİD3 THEN
EKSİİ
GOTO SANİ29
ELSE
GOTO DONE29
END IF
SANİ29:
FOR İ = 1 TO SAYİD3
IF A4$(İ) = "X" THEN

```

```

A4$(I) = STR$(-X10)
ELSE
GOTO MLKN29
END IF
MLKN29:
NEXT I
GOTO DONE29
SEAN29:
IF SAYID <= 1 AND F1$ <> "X" THEN
SND9 = SND9 + 1
SONUC3(10, SND9) = SA2(10, M, 1)
ARAL(10, M, 3) = 0
GOTO DONGU9
ELSE
GOTO SAN29
END IF
SAN29:
FOR I = 1 TO SAYID
IF A4$(I) = "X" THEN
A4$(I) = STR$(X10)
ELSE
GOTO MLK29
END IF
MLK29:
NEXT I
FOR I = 1 TO SAYID
FOR N = 1 TO K13(10)
IF A4$(I) = C13$(10, N) THEN
A4$(I) = STR$(SA2(10, M, N))
ELSE
GOTO FGH29
END IF
FGH29:
NEXT N
NEXT I
HESAPT
SND9 = SND9 + 1
SONUC3(10, SND9) = SONUC2
ARAL(10, M, 3) = 0
DONGU9:
NEXT M
END SELECT
'KILL "NUMBER2.DAT"
'KILL "HESAP.DAT"
OPEN "O", #18, "NUMBER2.DAT"
OPEN "O", #19, "HESAP.DAT"
WRITE #18, SND, SND1, SND2, SND3,
SND4, SND5, SND6, SND7, SND8, SND9
FOR K = 1 TO SND
WRITE #19, SONUC3(1, K)
NEXT K
FOR K = 1 TO SND1
WRITE #19, SONUC3(2, K)
NEXT K
FOR K = 1 TO SND2
WRITE #19, SONUC3(3, K)

```

```

NEXT K
FOR K = 1 TO SND3
WRITE #19, SONUC3(4, K)
NEXT K
FOR K = 1 TO SND4
WRITE #19, SONUC3(5, K)
NEXT K
FOR K = 1 TO SND5
WRITE #19, SONUC3(6, K)
NEXT K
FOR K = 1 TO SND6
WRITE #19, SONUC3(7, K)
NEXT K
FOR K = 1 TO SND7
WRITE #19, SONUC3(8, K)
NEXT K
FOR K = 1 TO SND8
WRITE #19, SONUC3(9, K)
NEXT K
FOR K = 1 TO SND9
WRITE #19, SONUC3(10, K)
NEXT K
END SUB
SUB HESAPT
ON PEN GOSUB PENREPORT
PRT = 0
PRA:
SAK = 1
PARANTEZ:
IF A3$(SAK) = "(" THEN
A3$(SAK) = ""
GOTO PISARETLER
ELSE
GOSUB ISA
GOTO PARANTEZ
END IF
ISA:
SAK = SAK + 1
IF SAK > ISARET THEN
GOTO DUZEN
ELSE
RETURN
END IF
PISARETLER:
ISARET1 = SAK
ISARET2 = SAK + 1
ISARET3 = 0
ISARL = 0
ISARET5 = ISARET
SAYID1 = SAYID
TEKA:
ISARET3 = ISARET3 + 1
A2$(ISARET3) = A3$(ISARET2)
ISARET2 = ISARET2 + 1
IF A3$(ISARET2) = ")" THEN
A3$(ISARET2) = ""

```

```

ISARET2 = ISARET2 + 1
GOTO DEGER1
ELSE
GOTO TEKA
END IF
DEGER1:
ISARET1 = ISARET1 - 1
FOR N = 1 TO ISARET3 + 1
ISARET1 = ISARET1 + 1
A1$(N) = A4$(ISARET1)
NEXT N
SB1 = ISARET3 + 1
SB2 = ISARET3
GOTO ISLEM
DUZEN:
KLL = 0
KLN = 0
FOR N = 1 TO ISARET
KLL = KLL + 1
A2$(KLL) = A3$(N)
NEXT N
FOR N = 1 TO SAYID
KLN = KLN + 1
A1$(KLN) = A4$(N)
NEXT N
SB1 = KLN
SB2 = KLL
KKK = 1
PRT = 1
ISARET = 0
FOR N = 1 TO KLL
A3$(N) = ""
NEXT N
FOR N = 1 TO KLN
A4$(N) = ""
NEXT N
ISLEM:
SA2 = SB2
SA1 = SB1
US:
I = 1
KN = 1
USSOR:
IF A2$(I) = "^" THEN
GOTO USALMA
ELSE
I = I + 1
GOSUB USSOR1
GOTO USSOR
END IF
USSOR1:
IF I <= SA2 THEN
RETURN
ELSE
GOTO BOL
END IF

```

```

USALMA:
I1 = I: I2 = I + 1: I3 = I + 2
I4 = I + 1: I5 = I + 2
KN1 = VAL(A1$(I1))
KN2 = VAL(A1$(I2))
F12$ = MID$(A1$(I1), 1, 1)
IF F12$ = "-" AND PRT = 0 AND KN2 = 2
THEN
SONUC(I) = -KN1 ^ KN2
ELSEIF F12$ = "-" AND PRT = 0 AND KN2
= 4 THEN
SONUC(I) = -KN1 ^ KN2
ELSEIF F12$ = "-" AND PRT = 0 AND KN2
= 6 THEN
SONUC(I) = -KN1 ^ KN2
ELSEIF F12$ = "-" AND PRT = 0 AND KN2
= 8 THEN
SONUC(I) = -KN1 ^ KN2
ELSE
SONUC(I) = KN1 ^ KN2
END IF
GOTO IVESY
BOL:
I = 1
KN = 2
BOLSOR:
IF A2$(I) = "/" THEN
GOTO BOLME
ELSE
I = I + 1
GOSUB BOLSOR1
GOTO BOLSOR
END IF
BOLSOR1:
IF I <= SA2 THEN
RETURN
ELSE
GOTO CAR
END IF
BOLME:
I1 = I: I2 = I + 1: I3 = I + 2
I4 = I + 1: I5 = I + 2
KN1 = VAL(A1$(I1))
KN2 = VAL(A1$(I2))
SONUC(I) = KN1 / KN2
GOTO IVESY
CAR:
I = 1
KN = 3
CARSOR:
IF A2$(I) = "*" THEN
GOTO CARPMA
ELSE
I = I + 1
GOSUB CARSOR1
GOTO CARSOR

```

```

END IF
CARSOR1:
IF I <= SA2 THEN
RETURN
ELSE
GOTO CIK
END IF
CARPMA:
I1 = I: I2 = I + 1: I3 = I + 2
I4 = I + 1: I5 = I + 2
KN1 = VAL(A1$(I1))
KN2 = VAL(A1$(I2))
SONUC(I) = KN1 * KN2
GOTO IVESY
CIK:
I = 1
KN = 4
CIKSOR:
IF A2$(I) = "-" THEN
GOTO CIKARMA
ELSE
I = I + 1
GOSUB CIKSOR1
GOTO CIKSOR
END IF
CIKSOR1:
IF I <= SA2 THEN
RETURN
ELSE
GOTO TOP
END IF
CIKARMA:
I1 = I: I2 = I + 1: I3 = I + 2: I4 = I + 1: I5 = I
+ 2
KN1 = VAL(A1$(I1))
KN2 = VAL(A1$(I2))
SONUC(I) = KN1 - KN2
GOTO IVESY
TOP:
I = 1
KN = 5
TOPSOR:
IF A2$(I) = "+" THEN
GOTO TOPLAMA
ELSE
I = I + 1
GOSUB TOPSOR1
GOTO TOPSOR
END IF
TOPSOR1:
IF I <= SA2 THEN
RETURN
ELSE
GOTO SONUC
END IF
TOPLAMA:

```

```

I1 = I: I2 = I + 1: I3 = I + 2
I4 = I + 1: I5 = I + 2
KN1 = VAL(A1$(I1))
KN2 = VAL(A1$(I2))
SONUC(I) = KN1 + KN2
GOTO IVESY
SOR:
IF SA2 >= 1 THEN
RETURN
ELSE
GOTO SONUC
END IF
IVESY:
IF I >= SA2 THEN
KN = KN + 1
GOTO BOS
ELSE
GOTO SN
END IF
SN:
I1 = I1 - 1
FOR N = I2 TO SA2
I1 = I1 + 1
A2$(I1) = A2$(N)
NEXT N
I4 = I4 - 1
FOR N = I5 TO SA1
I4 = I4 + 1
A1$(I4) = A1$(N)
NEXT N
BOS:
A1$(SA1) = ""
A2$(SA2) = ""
A1$(I) = STR$(SONUC(I))
SA1 = SA1 - 1
SA2 = SA2 - 1
GOSUB SOR
IF KN = 1 THEN
GOTO US
ELSEIF KN = 2 THEN
GOTO BOL
ELSEIF KN = 3 THEN
GOTO CAR
ELSEIF KN = 4 THEN
GOTO CIK
ELSE
GOTO TOP
END IF
SONUC:
SONUC2 = SONUC(I)

ATAMA:
IF ISARET2 > ISARET AND KKK = 1
THEN
GOTO SONUC1

```

```

ELSEIF ISARET2 > ISARET AND KKK <>
1 THEN
GOTO BOS1
ELSE
GOTO SN1
END IF
SN1:
ISARET4 = ISARET2
ISARET5 = SAK - 1
ISARET8 = ISARET
FOR N = ISARET4 TO ISARET8
ISARET5 = ISARET5 + 1
A3$(ISARET5) = A3$(N)
A3$(N) = ""
NEXT N
ISARET6 = SAK
ISARET7 = ISARET1 + 1
FOR N = ISARET7 TO SAYID1
ISARET6 = ISARET6 + 1
A4$(ISARET6) = A4$(N)
A4$(N) = ""
NEXT N
A4$(SAK) = STR$(SONUC2)
FOR N = 1 TO ISARET3 + 2
ISARET = ISARET - 1
NEXT N
FOR N = 1 TO ISARET3
SAYID = SAYID - 1
NEXT N
GOTO ASD
BOS1:
ISARET4 = SAK
ISARET8 = ISARET
FOR N = ISARET4 TO ISARET8
A3$(N) = ""
NEXT N
ISARET6 = SAK
FOR N = ISARET6 TO SAYID1
A4$(N) = ""
NEXT N
A4$(SAK) = STR$(SONUC2)
FOR N = 1 TO ISARET3 + 2
ISARET = ISARET - 1
NEXT N
FOR N = 1 TO ISARET3
SAYID = SAYID - 1
NEXT N
ASD:
IF ISARET >= 1 AND KKK <> 1 THEN
GOTO PRA
ELSE
GOTO SONUC1
END IF
END SUB
SUB KUMSEC

```

```

ON PEN GOSUB PENREPORT
OPEN "I", #7, "SINIR.DAT"
OPEN "I", #3, "SAYI.DAT"
OPEN "I", #2, "UKKAYIT.DAT"
FOR I = 1 TO 10
FOR N = 1 TO 7
FOR K = 1 TO 3
ARAL(I, N, K) = 0
NEXT K
NEXT N
NEXT I
HOGO:
SELECT CASE SAYI
CASE 1
REDIM GUK(1)
INPUT #3, GUK(1)
C1 = GUK(1)
REDIM GUK$(1, GUK(1))
REDIM ARA1(1, GUK(1), 2)
REDIM ARA2(1, GUK(1), 2)
REDIM ARA3(1, GUK(1), 2)
FOR K = 1 TO GUK(1)
INPUT #2, GUK$(1, K)
NEXT K
FOR I = 1 TO GUK(1)
FOR KL = 1 TO 2
INPUT #7, ARA1(1, I, KL)
NEXT KL
FOR KL = 1 TO 2
INPUT #7, ARA2(1, I, KL)
NEXT KL
FOR KL = 1 TO 2
INPUT #7, ARA3(1, I, KL)
NEXT KL
NEXT I
FOR I = 1 TO GUK(1)
IF X1 >= ARA1(1, I, 1) AND X1 <= A-
RA1(1, I, 2) AND (ARA1(1, I, 1) <> 0 OR
ARA1(1, I, 2) <> 0) THEN
ARAL(1, I, 1) = 1
ELSEIF X1 >= ARA2(1, I, 1) AND X1 <=
ARA2(1, I, 2) AND (ARA2(1, I, 1) <> 0 OR
ARA2(1, I, 2) <> 0) THEN
ARAL(1, I, 2) = 1
ELSEIF X1 >= ARA3(1, I, 1) AND X1 <=
ARA3(1, I, 2) AND (ARA3(1, I, 1) <> 0 OR
ARA3(1, I, 2) <> 0) THEN
ARAL(1, I, 3) = 1
ELSE
GOTO KN
END IF
KN:
NEXT I
ERASE ARA1
ERASE ARA2
ERASE ARA3

```

```

ERASE GUK
ERASE GUK$
SAYI = 2
GOTO HOGO
CASE 2
REDIM GUK(2 TO 2)
INPUT #3, GUK(2)
C2 = GUK(2)
REDIM GUK$(2 TO 2, GUK(2))
REDIM ARA1(2 TO 2, GUK(2), 2)
REDIM ARA2(2 TO 2, GUK(2), 2)
REDIM ARA3(2 TO 2, GUK(2), 2)
FOR K = 1 TO GUK(2)
INPUT #2, GUK$(2, K)
NEXT K
FOR I = 1 TO GUK(2)
FOR KL = 1 TO 2
INPUT #7, ARA1(2, I, KL)
NEXT KL
FOR KL = 1 TO 2
INPUT #7, ARA2(2, I, KL)
NEXT KL
FOR KL = 1 TO 2
INPUT #7, ARA3(2, I, KL)
NEXT KL
NEXT I
FOR I = 1 TO GUK(2)
IF X2 >= ARA1(2, I, 1) AND X2 <= A-
RA1(2, I, 2) AND (ARA1(2, I, 1) < 0 OR
ARA1(2, I, 2) < 0) THEN
ARAL(2, I, 1) = 1
ELSEIF X2 >= ARA2(2, I, 1) AND X2 <=
ARA2(2, I, 2) AND (ARA2(2, I, 1) < 0 OR
ARA2(2, I, 2) < 0) THEN
ARAL(2, I, 2) = 1
ELSEIF X2 >= ARA3(2, I, 1) AND X2 <=
ARA3(2, I, 2) AND (ARA3(2, I, 1) < 0 OR
ARA3(2, I, 2) < 0) THEN
ARAL(2, I, 3) = 1
ELSE GOTO KN1
END IF
KN1:
NEXT I
ERASE ARA1
ERASE ARA2
ERASE ARA3
ERASE GUK
ERASE GUK$
SAYI = 3
GOTO HOGO
CASE 3
REDIM GUK(3 TO 3)
INPUT #3, GUK(3)
C3 = GUK(3)
REDIM GUK$(3 TO 3, GUK(3))
REDIM ARA1(3 TO 3, GUK(3), 2)

```

```

REDIM ARA2(3 TO 3, GUK(3), 2)
REDIM ARA3(3 TO 3, GUK(3), 2)
FOR K = 1 TO GUK(3)
INPUT #2, GUK$(3, K)
NEXT K
FOR I = 1 TO GUK(3)
FOR KL = 1 TO 2
INPUT #7, ARA1(3, I, KL)
NEXT KL
FOR KL = 1 TO 2
INPUT #7, ARA2(3, I, KL)
NEXT KL
FOR KL = 1 TO 2
INPUT #7, ARA3(3, I, KL)
NEXT KL
NEXT I
FOR I = 1 TO GUK(3)
IF X3 >= ARA1(3, I, 1) AND X3 <= A-
RA1(3, I, 2) AND (ARA1(3, I, 1) < 0 OR
ARA1(3, I, 2) < 0) THEN
ARAL(3, I, 1) = 1
ELSEIF X3 >= ARA2(3, I, 1) AND X3 <=
ARA2(3, I, 2) AND (ARA2(3, I, 1) < 0 OR
ARA2(3, I, 2) < 0) THEN
ARAL(3, I, 2) = 1
ELSEIF X3 >= ARA3(3, I, 1) AND X3 <=
ARA3(3, I, 2) AND (ARA3(3, I, 1) < 0 OR
ARA3(3, I, 2) < 0) THEN
ARAL(3, I, 3) = 1
ELSE GOTO KN2
END IF
KN2:
NEXT I
ERASE ARA1
ERASE ARA2
ERASE ARA3
ERASE GUK
ERASE GUK$
SAYI = 4
GOTO HOGO
CASE 4
REDIM GUK(4 TO 4)
INPUT #3, GUK(4)
C4 = GUK(4)
REDIM GUK$(4 TO 4, GUK(4))
REDIM ARA1(4 TO 4, GUK(4), 2)
REDIM ARA2(4 TO 4, GUK(4), 2)
REDIM ARA3(4 TO 4, GUK(4), 2)
FOR K = 1 TO GUK(4)
INPUT #2, GUK$(4, K)
NEXT K
FOR I = 1 TO GUK(4)
FOR KL = 1 TO 2
INPUT #7, ARA1(4, I, KL)
NEXT KL
FOR KL = 1 TO 2

```



```

INPUT #7, ARA2(4, I, KL)
NEXT KL
FOR KL = 1 TO 2
INPUT #7, ARA3(4, I, KL)
NEXT KL
NEXT I
FOR I = 1 TO GUK(4)
IF X4 >= ARA1(4, I, 1) AND X4 <= A-
RA1(4, I, 2) AND (ARA1(4, I, 1) < 0 OR
ARA1(4, I, 2) < 0) THEN
ARAL(4, I, 1) = 1
ELSEIF X4 >= ARA2(4, I, 1) AND X4 <=
ARA2(4, I, 2) AND (ARA2(4, I, 1) < 0 OR
ARA2(4, I, 2) < 0) THEN
ARAL(4, I, 2) = 1
ELSEIF X4 >= ARA3(4, I, 1) AND X4 <=
ARA3(4, I, 2) AND (ARA3(4, I, 1) < 0 OR
ARA3(4, I, 2) < 0) THEN
ARAL(4, I, 3) = 1
ELSE GOTO KN3
END IF
KN3:
NEXT I
ERASE ARA1
ERASE ARA2
ERASE ARA3
ERASE GUK
ERASE GUK$
SAYI = 5
GOTO HOGO
CASE 5
REDIM GUK(5 TO 5)
INPUT #3, GUK(5)
C5 = GUK(5)
REDIM GUK$(5 TO 5, GUK(5))
REDIM ARA1(5 TO 5, GUK(5), 2)
REDIM ARA2(5 TO 5, GUK(5), 2)
REDIM ARA3(5 TO 5, GUK(5), 2)
FOR K = 1 TO GUK(5)
INPUT #2, GUK$(5, K)
NEXT K
FOR I = 1 TO GUK(5)
FOR KL = 1 TO 2
INPUT #7, ARA1(5, I, KL)
NEXT KL
FOR KL = 1 TO 2
INPUT #7, ARA2(5, I, KL)
NEXT KL
FOR KL = 1 TO 2
INPUT #7, ARA3(5, I, KL)
NEXT KL
NEXT I
FOR I = 1 TO GUK(5)
IF X5 >= ARA1(5, I, 1) AND X5 <= A-
RA1(5, I, 2) AND (ARA1(5, I, 1) < 0 OR
ARA1(5, I, 2) < 0) THEN

```

```

ARAL(5, I, 1) = 1
ELSEIF X5 >= ARA2(5, I, 1) AND X5 <=
ARA2(5, I, 2) AND (ARA2(5, I, 1) < 0 OR
ARA2(5, I, 2) < 0) THEN
ARAL(5, I, 2) = 1
ELSEIF X5 >= ARA3(5, I, 1) AND X5 <=
ARA3(5, I, 2) AND (ARA3(5, I, 1) < 0 OR
ARA3(5, I, 2) < 0) THEN
ARAL(5, I, 3) = 1
ELSE GOTO KN4
END IF
KN4:
NEXT I
ERASE ARA1
ERASE ARA2
ERASE ARA3
ERASE GUK
ERASE GUK$
SAYI = 6
GOTO HOGO
CASE 6
REDIM GUK(6 TO 6)
INPUT #3, GUK(6)
C6 = GUK(6)
REDIM GUK$(6 TO 6, GUK(6))
REDIM ARA1(6 TO 6, GUK(6), 2)
REDIM ARA2(6 TO 6, GUK(6), 2)
REDIM ARA3(6 TO 6, GUK(6), 2)
FOR K = 1 TO GUK(6)
INPUT #2, GUK$(6, K)
NEXT K
FOR I = 1 TO GUK(6)
FOR KL = 1 TO 2
INPUT #7, ARA1(6, I, KL)
NEXT KL
FOR KL = 1 TO 2
INPUT #7, ARA2(6, I, KL)
NEXT KL
FOR KL = 1 TO 2
INPUT #7, ARA3(6, I, KL)
NEXT KL
NEXT I
FOR I = 1 TO GUK(6)
IF X6 >= ARA1(6, I, 1) AND X6 <= A-
RA1(6, I, 2) AND (ARA1(6, I, 1) < 0 OR
ARA1(6, I, 2) < 0) THEN
ARAL(6, I, 1) = 1
ELSEIF X6 >= ARA2(6, I, 1) AND X6 <=
ARA2(6, I, 2) AND (ARA2(6, I, 1) < 0 OR
ARA2(6, I, 2) < 0) THEN
ARAL(6, I, 2) = 1
ELSEIF X6 >= ARA3(6, I, 1) AND X6 <=
ARA3(6, I, 2) AND (ARA3(6, I, 1) < 0 OR
ARA3(6, I, 2) < 0) THEN
ARAL(6, I, 3) = 1
ELSE

```

```

GOTO KN5
END IF
KN5:
NEXT I
ERASE ARA1
ERASE ARA2
ERASE ARA3
ERASE GUK
ERASE GUK$
SAYI = 7
GOTO HOGO
CASE 7
REDIM GUK(7 TO 7)
INPUT #3, GUK(7)
C7 = GUK(7)
REDIM GUK$(7 TO 7, GUK(7))
REDIM ARA1(7 TO 7, GUK(7), 2)
REDIM ARA2(7 TO 7, GUK(7), 2)
REDIM ARA3(7 TO 7, GUK(7), 2)
FOR K = 1 TO GUK(7)
INPUT #2, GUK$(7, K)
NEXT K
FOR I = 1 TO GUK(7)
FOR KL = 1 TO 2
INPUT #7, ARA1(7, I, KL)
NEXT KL
FOR KL = 1 TO 2
INPUT #7, ARA2(7, I, KL)
NEXT KL
FOR KL = 1 TO 2
INPUT #7, ARA3(7, I, KL)
NEXT KL
NEXT I
FOR I = 1 TO GUK(7)
IF X7 >= ARA1(7, I, 1) AND X7 <= A-
RA1(7, I, 2) AND (ARA1(7, I, 1) < 0 OR
ARA1(7, I, 2) < 0) THEN
ARAL(7, I, 1) = 1
ELSEIF X7 >= ARA2(7, I, 1) AND X7 <=
ARA2(7, I, 2) AND (ARA2(7, I, 1) < 0 OR
ARA2(7, I, 2) < 0) THEN
ARAL(7, I, 2) = 1
ELSEIF X7 >= ARA3(7, I, 1) AND X7 <=
ARA3(7, I, 2) AND (ARA3(7, I, 1) < 0 OR
ARA3(7, I, 2) < 0) THEN
ARAL(7, I, 3) = 1
ELSE GOTO KN6
END IF
KN6:
NEXT I
ERASE ARA1
ERASE ARA2
ERASE ARA3
ERASE GUK
ERASE GUK$
SAYI = 8

```

```

GOTO HOGO
CASE 8
REDIM GUK(8 TO 8)
INPUT #3, GUK(8)
C8 = GUK(8)
REDIM GUK$(8 TO 8, GUK(8))
REDIM ARA1(8 TO 8, GUK(8), 2)
REDIM ARA2(8 TO 8, GUK(8), 2)
REDIM ARA3(8 TO 8, GUK(8), 2)
FOR K = 1 TO GUK(8)
INPUT #2, GUK$(8, K)
NEXT K
FOR I = 1 TO GUK(8)
FOR KL = 1 TO 2
INPUT #7, ARA1(8, I, KL)
NEXT KL
FOR KL = 1 TO 2
INPUT #7, ARA2(8, I, KL)
NEXT KL
FOR KL = 1 TO 2
INPUT #7, ARA3(8, I, KL)
NEXT KL
NEXT I
FOR I = 1 TO GUK(8)
IF X8 >= ARA1(8, I, 1) AND X8 <= A-
RA1(8, I, 2) AND (ARA1(8, I, 1) < 0 OR
ARA1(8, I, 2) < 0) THEN
ARAL(8, I, 1) = 1
ELSEIF X8 >= ARA2(8, I, 1) AND X8 <=
ARA2(8, I, 2) AND (ARA2(8, I, 1) < 0 OR
ARA2(8, I, 2) < 0) THEN
ARAL(8, I, 2) = 1
ELSEIF X8 >= ARA3(8, I, 1) AND X8 <=
ARA3(8, I, 2) AND (ARA3(8, I, 1) < 0 OR
ARA3(8, I, 2) < 0) THEN
ARAL(8, I, 3) = 1
ELSE GOTO KN7
END IF
KN7:
NEXT I
ERASE ARA1
ERASE ARA2
ERASE ARA3
ERASE GUK
ERASE GUK$
SAYI = 9
GOTO HOGO
CASE 9
REDIM GUK(9 TO 9)
INPUT #3, GUK(9)
C9 = GUK(9)
REDIM GUK$(9 TO 9, GUK(9))
REDIM ARA1(9 TO 9, GUK(9), 2)
REDIM ARA2(9 TO 9, GUK(9), 2)
REDIM ARA3(9 TO 9, GUK(9), 2)
FOR K = 1 TO GUK(9)

```

```

INPUT #2, GUK$(9, K)
NEXT K
FOR I = 1 TO GUK(9)
FOR KL = 1 TO 2
INPUT #7, ARA1(9, I, KL)
NEXT KL
FOR KL = 1 TO 2
INPUT #7, ARA2(9, I, KL)
NEXT KL
FOR KL = 1 TO 2
INPUT #7, ARA3(9, I, KL)
NEXT KL
NEXT I
FOR I = 1 TO GUK(9)
IF X9 >= ARA1(9, I, 1) AND X9 <= A-
RA1(9, I, 2) AND (ARA1(9, I, 1) < 0 OR
ARA1(9, I, 2) < 0) THEN
ARAL(9, I, 1) = 1
ELSEIF X9 >= ARA2(9, I, 1) AND X9 <=
ARA2(9, I, 2) AND (ARA2(9, I, 1) < 0 OR
ARA2(9, I, 2) < 0) THEN
ARAL(9, I, 2) = 1
ELSEIF X9 >= ARA3(9, I, 1) AND X9 <=
ARA3(9, I, 2) AND (ARA3(9, I, 1) < 0 OR
ARA3(9, I, 2) < 0) THEN
ARAL(9, I, 3) = 1
ELSE GOTO KN8
END IF
KN8:
NEXT I
ERASE ARA1
ERASE ARA2
ERASE ARA3
ERASE GUK
ERASE GUK$
SAYI = 10
GOTO HOGO
CASE 10
REDIM GUK(10 TO 10)
INPUT #3, GUK(10)
C10 = GUK(10)
REDIM GUK$(10 TO 10, GUK(10))
REDIM ARA1(10 TO 10,
GUK(10), 2)
REDIM ARA2(10 TO 10,
GUK(10), 2)
REDIM ARA3(10 TO 10,
GUK(10), 2)
FOR K = 1 TO GUK(10)
INPUT #2, GUK$(10, K)
NEXT K
FOR I = 1 TO GUK(10)
FOR KL = 1 TO 2
INPUT #7, ARA1(10, I, KL)
NEXT KL
FOR KL = 1 TO 2

```

```

INPUT #7, ARA2(10, I, KL)
NEXT KL
FOR KL = 1 TO 2
INPUT #7, ARA3(10, I, KL)
NEXT KL
NEXT I
FOR I = 1 TO GUK(10)
IF X10 >= ARA1(10, I, 1) AND X10 <= A-
RA1(10, I, 2) AND (ARA1(10, I, 1) < 0 OR
ARA1(10, I, 2) < 0) THEN
ARAL(10, I, 1) = 1
ELSEIF X10 >= ARA2(10, I, 1) AND X10 <=
ARA2(10, I, 2) AND (ARA2(10, I, 1) < 0
OR ARA2(10, I, 2) < 0) THEN
ARAL(10, I, 2) = 1
ELSEIF X10 >= ARA3(10, I, 1) AND X10 <=
ARA3(10, I, 2) AND (ARA3(10, I, 1) < 0
OR ARA3(10, I, 2) < 0) THEN
ARAL(10, I, 3) = 1
ELSE GOTO KN9
END IF
KN9:
NEXT I
ERASE ARA1
ERASE ARA2
ERASE ARA3
ERASE GUK
ERASE GUK$
CLOSE #2, #3, #7
END SELECT
OPEN "O", #17, "SUKK.DAT"
FOR K = 1 TO C1
FOR N = 1 TO 3
WRITE #17, ARAL(1, K, N)
NEXT N
NEXT K
FOR K = 1 TO C2
FOR N = 1 TO 3
WRITE #17, ARAL(2, K, N)
NEXT N
NEXT K
FOR K = 1 TO C3
FOR N = 1 TO 3
WRITE #17, ARAL(3, K, N)
NEXT N
NEXT K
FOR K = 1 TO C4
FOR N = 1 TO 3
WRITE #17, ARAL(4, K, N)
NEXT N
NEXT K
FOR K = 1 TO C5
FOR N = 1 TO 3
WRITE #17, ARAL(5, K, N)
NEXT N
NEXT K

```

```

FOR K = 1 TO C6
FOR N = 1 TO 3
WRITE #17, ARAL(6, K, N)
NEXT N
NEXT K
FOR K = 1 TO C7
FOR N = 1 TO 3
WRITE #17, ARAL(7, K, N)
NEXT N
NEXT K
FOR K = 1 TO C8
FOR N = 1 TO 3
WRITE #17, ARAL(8, K, N)
NEXT N
NEXT K
FOR K = 1 TO C9
FOR N = 1 TO 3
WRITE #17, ARAL(9, K, N)
NEXT N
NEXT K
FOR K = 1 TO C10
FOR N = 1 TO 3
WRITE #17, ARAL(10, K, N)
NEXT N
NEXT K
CLOSE #17
END SUB
SUB PORT
ARALIK = 18 / 4096
NIN VOLTAJ DEGERI
FOR CHANEL = 0 TO 9
OUT &H278, CHANEL
OUT &H27B, CHANEL
FOR I = 1 TO 5
A = INP(&H27C)
NEXT
FOR I = 1 TO 9
B = INP(&H27D)
NEXT
C = INP(&H27A)
D = INP(&H279)
VERI = ((C AND &HF) * 256) + D
IF VERI > 2048 THEN DATI = VERI - 2048
ELSEIF VERI = 2048 THEN DATI = 0 ELSEIF VERI < 2048 THEN DATI = -(2048 - VERI)
VOLTAJ = DATI * ARALIK
X(CHANEL + 1) = VOLTAJ
NEXT
X(2) = 60
SAYI = 1
X1 = X(1)
X2 = X(2)
X3 = X(3)
X4 = X(4)
X5 = X(5)

```

HER BİR SAYI-

```

X6 = X(6)
X7 = X(7)
X8 = X(8)
X9 = X(9)
X10 = X(10)
FOR I = 1 TO 10
LOCATE 4 + I, 60
PRINT I; "GİRİŞİ="; X(I)
NEXT
END SUB

```

***** KURALS3*****

```

DECLARE SUB AKTIFC ()
DECLARE SUB GRAPHMAKE (PORT!)
DECLARE SUB MIN ()
DECLARE SUB SECK () DECLARE SUB
CIKB ()
DECLARE SUB AKTIFC ()
COMMON SHARED Q, D2, D1, CUK, SONUC1, SONUC2, PORTC
DIM SHARED INREGS AS REGTYPE,
OUTREGS AS REGTYPE
INREGS.AX = &H1
CALL INTERRUPT(&H33, INREGS, OUTREGS)
PEN ON 'ENABLE LIGHT PEN.
INREGS.AX = &HD
CALL INTERRUPT(&H33, INREGS, OUTREGS)
TIMER ON
ON TIMER(1) GOSUB PENREPORT
REDIM SHARED SONUC3(10, 1 TO 7)
REDIM SHARED SONUC4(360)
REDIM SHARED GUK(10)
REDIM SHARED KURS(360)
REDIM SHARED ARAL(10, 1 TO 7, 3)
REDIM SHARED KURS$(360)
REDIM SHARED CUK$(1 TO 7)
REDIM SHARED SAC(1, 1 TO 7)
CLOSE
AKTIFC
CIKB
MIN
GRAPHMAKE PORTC
CLEAR
CHAIN "COZUM12"
GRAPHFILE
CLOSE
OPEN "O", #1, "GRAPHMAK.FLM"
WRITE #1, 0, 0, 0, 0
CLOSE
RESUME NEXT
PENREPORT
AD$ = INKEY$
IF AD$ = CHR$(27) THEN END

```

```

IF AD$ = CHR$(0) + "," THEN Y = 135: X =
68
GOTO A
IF AD$ = CHR$(0) + "<" THEN Y = 135
X = 141
GOTO A
IF AD$ = CHR$(0) + "=" THEN Y = 135
X = 213
GOTO A
IF AD$ = CHR$(0) + ">" THEN Y = 135
X = 285
GOTO A
IF -1 = PEN(3) THEN
'REPORT LIGHT PEN STATUS AND GET
X AND Y POSITION.
X = PEN(4)
Y = PEN(5)
A: IF Y >= 132 AND Y <= 155 THEN
IF X >= 47 AND X <= 135 THEN PEN OFF
INREGS.AX = &HE
CALL INTERRUPT(&H33, INREGS, OUT-
REGS)
CLS
SCREEN 0
END
IF X >= 140 AND X <= 206 THEN PEN OFF
INREGS.AX = &HE
CALL INTERRUPT(&H33, INREGS, OUT-
REGS)
RUN "36A"
IF X >= 212 AND X <= 279 THEN DO
K = 1
IF PEN(0) = -1 THEN
X = PEN(4): Y = PEN(5)
IF Y >= 132 AND Y <= 155 THEN
IF X >= 67 AND X <= 135 THEN PEN OFF
INREGS.AX = &HE
CALL INTERRUPT(&H33, INREGS, OUT-
REGS)
CLS
SCREEN 0
END
IF X >= 140 AND X <= 206 THEN PEN OFF
INREGS.AX = &HE
CALL INTERRUPT(&H33, INREGS, O-
UTREGS)
RUN "36A"
IF X >= 284 AND X <= 351 THEN K = 0
END IF
END IF
LOOP UNTIL K = 0
END IF
IF X >= 357 AND X <= 424 THEN K = 0
END IF
END IF
RETURN

```

```

SUB AKTIFC
ON PEN GOSUB PENREPORT
OPEN "I", #12, "KURALL.DAT"
OPEN "I", #17, "SUKK.DAT"
OPEN "I", #3, "SAYI.DAT"
OPEN "I", #2, "UKKAYIT.DAT"
OPEN "I", #5, "DEGER1.DAT"
REDIM K11(10)
REDIM K12(10)
REDIM K13(10)
REDIM SA(10, 1 TO 7, 2)
REDIM SA1(10, 1 TO 7, 2)
REDIM SA2(10, 1 TO 7, 2)
REDIM GUK$(10, 1 TO 7)
FOR I = 1 TO 10
INPUT #5, K11(I), K12(I), K13(I)
NEXT I
CLOSE #5
FOR I = 1 TO 10
INPUT #3, GUK(I)
NEXT I
INPUT #3, CUK
CLOSE #3
FOR I = 1 TO 10
FOR M = 1 TO GUK(I)
INPUT #2, GUK$(I, GUK(I))
NEXT M
NEXT I
FOR K = 1 TO CUK
INPUT #2, CUK$(K)
NEXT K
CLOSE #2
FOR I = 1 TO 10
FOR KR = 1 TO GUK(I)
FOR KR1 = 1 TO 3
INPUT #17, ARAL(I, KR, KR1)
NEXT KR1
NEXT KR
NEXT I
CLOSE #17
KP = 1
KP1 = 2
KP4 = 1
D1 = 0
D2 = 0
DFG:
KP2 = GUK(KP) * GUK(KP1)
REDIM KURAL$(KP TO KP, KP1 TO KP1,
KP2)
FOR I = 1 TO KP2
INPUT #12, KURAL$(KP, KP1, I)
NEXT I
FOR KR = 1 TO GUK(KP)
IF ARAL(KP, KR, 1) = 1 OR ARAL(KP, KR,
2) = 1 OR ARAL(KP, KR, 3) = 1 THEN
GOSUB SORI

```

EK-1

```

KP4 = (KP2 - (GUK(KP) - 1) * GUK(KP1) +
1)
ELSE
KP4 = (KP2 - (GUK(KP) - 1) * GUK(KP1) +
1)
END IF
NEXT KR
ERASE KURAL$
KP4 = 1
KP1 = KP1 + 1
IF KP1 <= 10 THEN
GOTO DFG
ELSE
GOTO SOR2
END IF
SOR1:
FOR KL = 1 TO GUK(KP1)
IF ARAL(KP1, KL, 1) = 1 OR ARAL(KP1,
KL, 2) = 1 OR ARAL(KP1, KL, 3) = 1 THEN
D1 = D1 + 1
KURS$(D1) = KURAL$(KP, KP1, KP4)
KP4 = KP4 + 1
ELSE
KP4 = KP4 + 1
END IF
NEXT KL
RETURN
SOR2:
KP = KP + 1
KP1 = KP + 1
IF KP <= 9 THEN
GOTO DFG
ELSE
CLOSE #12
ERASE KURAL$
END IF
OPEN "I", #6, "SABITGC.DAT"
FOR I = 1 TO 10
FOR K = 1 TO GUK(I)
FOR L = 1 TO K11(I)
INPUT #6, SA(I, K, L)
NEXT L
FOR L = 1 TO K12(I)
INPUT #6, SA1(I, K, L)
NEXT L
FOR L = 1 TO K13(I)
INPUT #6, SA2(I, K, L)
NEXT L
NEXT K
NEXT I
FOR I = 1 TO CUK
INPUT #6, SAC(1, I)
NEXT I
CLOSE #6
ERASE K11
ERASE K12

```

```

ERASE K13
ERASE GUK$
ERASE SA
ERASE SA1
ERASE SA2
END SUB
SUB CIKB
ON PEN GOSUB PENREPORT
FOR I = 1 TO D1
FOR K = 1 TO CUK
IF KURS$(I) = CUK$(K) THEN
D2 = D2 + 1
KURS(D2) = SAC(1, K)
ELSE
GOTO DONG
END IF
DONG:
NEXT K
NEXT I
END SUB
SUB GRAPHMAKE (PORT)
ON PEN GOSUB PENREPORT
ON ERROR GOTO GRAPHFILE
ARALIK = 18 / 4096 'HER BİR SAYI-
NİN VOLTAJ DEĞERİ
CHANEL = 10
OUT &H278, CHANEL
OUT &H27B, CHANEL
FOR I = 1 TO 5
A = INP(&H27C)
NEXT
FOR I = 1 TO 9
B = INP(&H27D)
NEXT
C = INP(&H27A)
D = INP(&H279)
VERİ = ((C AND &HF) * 256) + D
IF VERİ > 2048 THEN DATİ = VERİ - 2048
ELSE IF VERİ = 2048 THEN DATİ = 0 EL-
SE IF VERİ < 2048 THEN DATİ = -(2048 -
VERİ)
VOLTAJ = DATİ * ARALIK
OPEN "I", #1, "GRAPHMAK.FLM"
INPUT #1, PORT0, TIME, PORT1, TEORİK
CLOSE
DPORT = (TEORİK - PORT) / 1.8
NTIME = TIME + 1
TANIMI
VIEW (55, 45)-(445, 345)
WINDOW SCREEN (0, 0)-(300, 180)
IF NTIME = 301 THEN NTIME = 0 TIME =
0
LINE (0, 0)-(300, 180), 8, BF
LINE (0, 90)-(300, 90), 14
LINE (TIME, 90 - PORT0)-(NTIME, 90 -
DPORT), 15

```

```

LINE (TIME, 90 - PORT1)-(NTIME, 90 -
PORT), 0
LOCATE 16, 60
PRINT "TEORİK ="; TEORİK
LOCATE 15, 60
PRINT "ORTAM ="; VOLTAJ
LOCATE 17, 60
PRINT "ÇIKIŞ ="; PORT
OPEN "O", #1, "GRAPHMAK.FLM"
WRITE #1, DPORT, NTIME, PORT, TEO-
RİK
CLOSE
END SUB
SUB MIN
ON PEN GOSUB PENREPORT
OPEN "I", #18, "NUMBER2.DAT"
INPUT #18, SND, SND1, SND2, SND3,
SND4, SND5, SND6, SND7, SND8, SND9
CLOSE #18
REDIM SON(10)
SON(1) = SND: SON(2) = SND1 SON(3) =
SND2: SON(4) = SND3 SON(5) = SND4
SON(6) = SND5: SON(7) = SND6 SON(8) =
SND7: SON(9) = SND8 SON(10) = SND9
OPEN "I", #19, "HESAP.DAT"
FOR İ = 1 TO 10
FOR K = 1 TO SON(İ)
INPUT #19, SONUC3(İ, K)
TOPLAM = TOPLAM + SONUC3(İ, K)
NEXT K
NEXT İ
KP6 = 1
KP7 = 2
DFR = 0
DFR1 = 0
DFG2:
FOR KR = 1 TO GUK(KP6)
IF ARAL(KP6, KR, 1) = 1 OR ARAL(KP6,
KR, 2) = 1 OR ARAL(KP6, KR, 3) = 1 THEN
DFR = DFR + 1
SONUC1 = SONUC3(KP6, DFR)
GOSUB SOR4
ELSE
END IF
NEXT KR
DFR = 0
KP7 = KP7 + 1
IF KP7 <= 10 THEN
GOTO DFG2
ELSE
GOTO SOR5
END IF
SOR4:
FOR KL = 1 TO GUK(KP7)
IF ARAL(KP7, KL, 1) = 1 OR ARAL(KP7,
KL, 2) = 1 OR ARAL(KP7, KL, 3) = 1 THEN

```

```

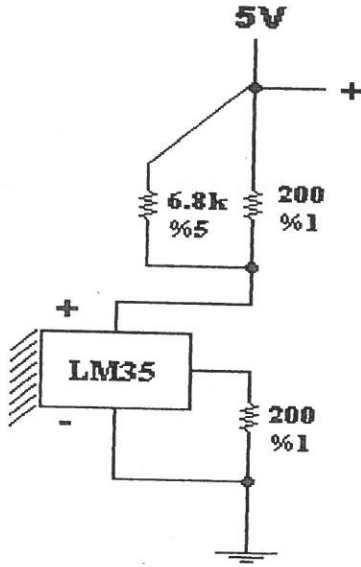
DFR1 = DFR1 + 1
SONUC2 = SONUC3(KP7, DFR1)
SECK
ELSE
END IF
NEXT KL
DFR1 = 0
RETURN
SOR5:
KP6 = KP6 + 1
KP7 = KP6 + 1
IF KP6 <= 9 THEN
GOTO DFG2
ELSE
END IF
ERASE SON
FOR İ = 1 TO Q
CARP = KURS(İ) * SONUC4(İ)
CARPC = CARPC + CARP
NEXT İ
IF TOPLAM = 0 THEN
CARPC = 0
EXIT SUB
ELSE
PORTC = CARPC / TOPLAM
PORTC = PORTC * 10 ^ 7
END IF
END SUB
SUB SECK
ON PEN GOSUB PENREPORT
IF SONUC1 < SONUC2 THEN
Q = Q + 1
SONUC4(Q) = SONUC1
ELSEIF SONUC1 = SONUC2 THEN
Q = Q + 1
SONUC4(Q) = SONUC1
ELSE
Q = Q + 1
SONUC4(Q) = SONUC2
END IF
END SUB

```

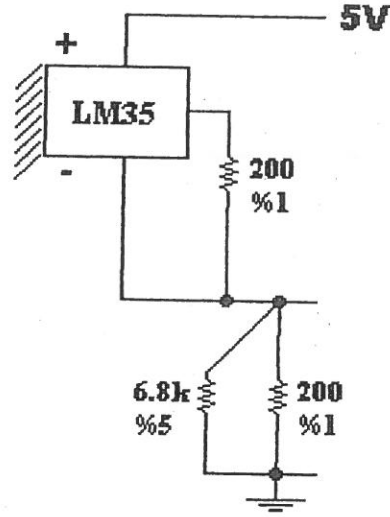
YÜKSEK HASSASİYETLİ DERECE CİNSİNDEN ISI ÖLÇER
(LM35 / LM35A/ LM35C/ LM35CA/ LM35D)

LM 35 serisi entegreler, derece cinsinden yüksek hassasiyette ısı ölçen algılatıcılardır. Çıkış voltajları, lineer bir şekilde ısı ölçü birimi santigrat (Centigrate) ile orantılıdır. Bundan dolayı LM 35 serisi entegrelerde, Kelvin cinsinden çıkış veren entegrelerde olduğu gibi kalibrasyona gerek kalmaz. Kullanıcının santigrat cinsinin uygun skalayı yakalaması için ara devreler kullanılmasına gerek kalmaz. Bu serinin ölçüm aralığı - 55 ile +150 °C arasındadır. Oda sıcaklığında $\pm 1/4$ °C doğrulukta ölçüm yapabilirler. LM 35'in alçak çıkış empedansı, lineer çıkış voltajı, otomatik olarak yapılan kalibrasyonun, onu kullanışlı bir sensör durumuna getirmiştir. Tek bir güç kaynağı ile normal veya simetrik bir besleme ile çalıştırılabilirler. Çalışma anında güç kaynağından 60 μ A gibi küçük bir akım çektiğinden, 0.1°C' den daha az ısınırlar. LM 35 entegresinin belli başlı özellikleri aşağıdaki gibi verilmiştir.

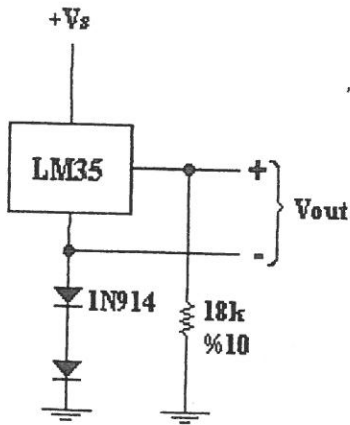
- 1) Doğrudan santigrat cinsinden ölçüm yaparlar.
- 2) + 10 mV/°C skala faktöründe lineerdir.
- 3) -55 ile +150 °C arasında ölçüm yapabilirler.
- 4) Düşük maliyetlidirler.
- 5) 4V ile 30V arasında bir besleme ile çalışırlar.
- 6) 60 μ A' den az akım çekerler.
- 7) Çalışma anında 0.08 °C' den daha az ısınırlar.
- 8) Nonlineeritesi $\pm 1/4$ °C 'dir.
- 9) Çıkış empedansı oldukça düşüktür.



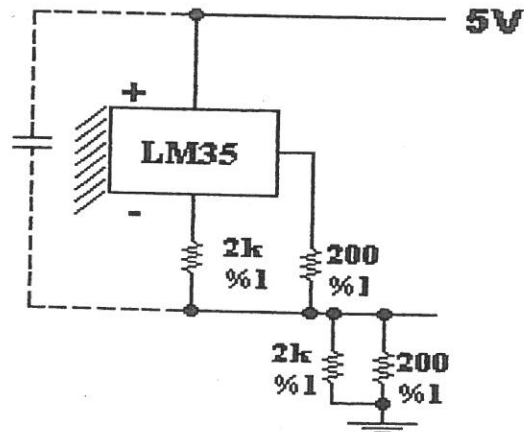
Şekil Ek-2.1. Ölçüm aralığı $+2^{\circ}\text{C}$ ile $+40^{\circ}\text{C}$ olan devre



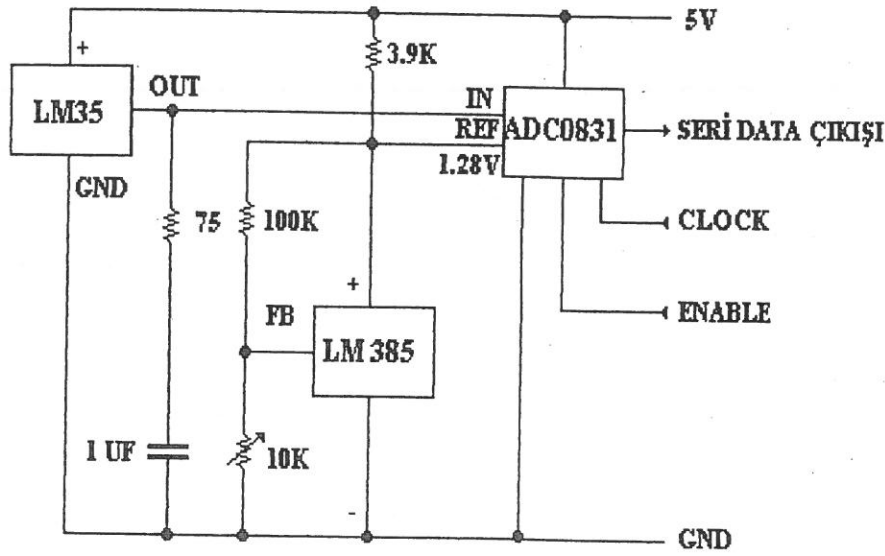
Şekil Ek-2.2. Ölçüm aralığı $+2^{\circ}\text{C}$ ile $+40^{\circ}\text{C}$ olan devre.



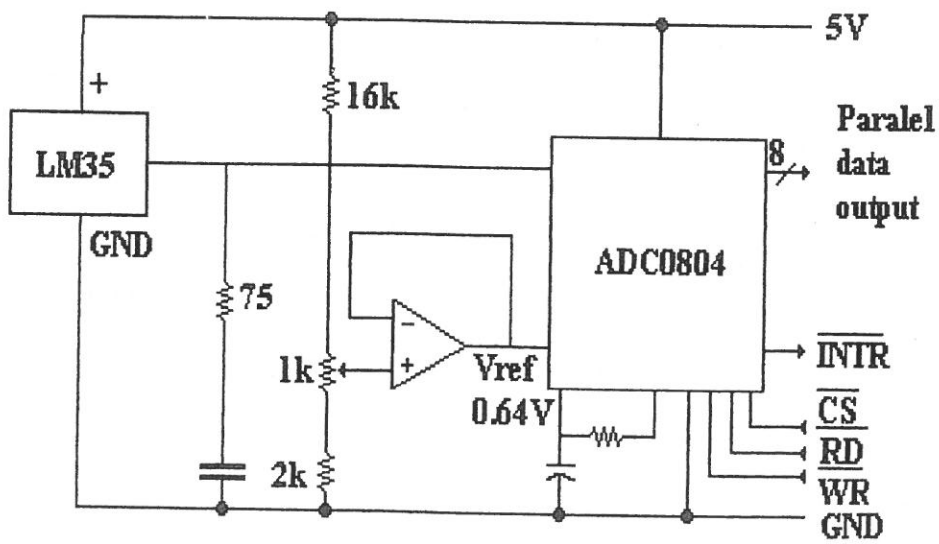
Şekil Ek-2.3. Ölçüm aralığı 55° ile $+150^{\circ}\text{C}$ olan devre



Şekil Ek-2.4. Ölçüm aralığı -5°C ile $+40^{\circ}\text{C}$ olan devre.



Şekil Ek-2.9. Dijital çevirici kullanarak yapılan sıcaklık ölçer.



Şekil Ek-2.10. Standart veri yollarına sahip mikro işlemciler için geliştirilmiş dijital ısı çevirici.

ÖZGEÇMİŞ

İsmail Atacak 1972 yılında Konya' nın Doğanhisar ilçesinde doğdu. İlkokul ve orta okulu Ayaslar Kasabasında bitirdi. 1990 yılında Gazi Üniversitesi Teknik Eğitim Fakültesi Elektronik-Bilgisayar Eğitimi Bölümü'ne girdi, 1994 yılında mezun oldu.

Aynı yıl Gazi Üniversitesi Fen Bilimleri Enstitüsünde yüksek lisansa başladı. 1996 yılında Gazi Üniversitesi Teknik Eğitim Fakültesi'ne Araştırma Görevlisi olarak girdi.