

# **C/C++ Programlama Dili**

**Ankara Üniversitesi, Mühendislik Fakültesi**

**Fizik Mühendisliği Bölümü**

**FZM206 Bilgisayar Programlama - II**

**C/C++**

**Hazırlayan : Mehmet Kabak**

**Nisan 2017**



Bu notların hazırlanmasındaki amaç, okuyucuya bilgisayarın programlamasının yapılabilme yeteneğinin kazandırılmasıdır. Burada C/C++ programlama dilinin çok fazla ayrıntısına girilmeden bilgisayar programlamasının öğrenilmesi hedeflenmiştir. Problemleri çözebilmek için fizik ve matematik bilginizin üniversite fiziğine giriş dersleri seviyesinde olması yeterli olacaktır. Program algoritmaları kullanılarak diğer programlama dillerinde aynı problemi yeniden yazabilirsiniz. Buradaki notlara haftada ortalama 4 saat zaman ayırırsanız 2-3 ay içerisinde birçok konuyu kavramış olursunuz.

İyi çalışmalar dilerim.

Mehmet KABAK

C/C++ Ders Notları-A.Ü. Fizik Mühendisliği Bölümü

## İÇİNDEKİLER

ŞEKİLLER DİZİNİ.....	vi
ALGORİTMALAR DİZİNİ.....	vii
PROGRAMLAR DİZİNİ.....	viii
ÇİZELGELER DİZİNİ .....	x
1. BÖLÜM (1nci ve 2nci haftalar).....	1
1.1 YAZILIM MÜHENDİSLİĞİ ve C++ .....	1
1.2 ALGORİTMA ve AKIŞ DİYAGRAMLARI.....	2
1.3 SORULAR.....	7
2. BÖLÜM (3ncü ve 4ncü haftalar).....	9
2.1 VERİ TİPLERİ (Data Types).....	9
2.2 DEĞİŞKEN veya DEĞİŞMEZLERİN İSİMLENDİRME KURALLARI .....	10
2.3 TAMSAYILAR, GERÇEL SAYILAR ve KARAKTERLER.....	11
2.4 ARİTMETİK İŞLEMCİLER.....	13
2.4.1 Aritmetik İşlemlerde Öncelik Sırası.....	13
2.4.2 Artırma ++ ve Azaltma -- İşlemleri .....	14
2.4.3 Karşılaştırma İşlemleri .....	17
2.5 ÖNİŞLEMCI (HEADER) KOMUTLARI.....	17
2.5.1 C++ Açıklama Satırları .....	19
2.5.2 Blok.....	19
2.6 VERİ DÖNÜŞÜMÜ.....	20
2.7 SORULAR.....	22
3. VERİ GİRİŞ-ÇIKIŞI (5nci, 6ncı ve 7nci haftalar).....	23
3.1 printf () ve cout FONKSİYONLARI.....	23
3.2 if Deyimi.....	32
3.3 goto Deyimi .....	33
3.4 Switch-case Deyimi.....	34
3.5 Dosyalar.....	38
3.6 Karakter Katarları (strings).....	41
3.7 SORULAR.....	42
4. DÖNGÜLER (8nci ve 9ncü haftalar).....	45
4.1 for DÖNGÜSÜ.....	45
4.1.1 İç içe for döngülerinin kullanılması .....	46
4.2 while Döngüsü.....	49
4.3 do while Döngüsü.....	49
4.3.1 İç içe do ... while döngülerin kullanılması. ....	50
4.4 SORULAR.....	59
5. BÖLÜM (10ncü ve 11nci haftalar).....	61
5.1 Fonksiyon Alt Programları .....	61
5.2 SORULAR.....	69
6. DİZİLER (arrays) (12nci ve 13ncü haftalar).....	71
6.1 Dizilerin Tanımlanması .....	71
6.2 SORULAR.....	81

7.	VERİ YAPILARI (data stuctures) (14ncü hafta).....	83
7.1	Yapısal Elemanlara Değer Atanması.....	87
7.2	İşaretçiler-Pointers.....	89
7.3	Sınıflar .....	91
7.4	SORULAR.....	95
8.	EK 1. Dev-C++ nin kullanımı.....	97

C/C++ Ders Notları-A.Ü. Fizik Mühendisliği Bölümü

## ŞEKİLLER DİZİNİ

Şekil 1-1 İkinci dereceden bir polinomun köklerinin hesaplanmasının akış diyagramı. ....	3
Şekil 1-2 Örnek 2-1 için iki sayının toplamını yapan akış diyagramı. ....	3
Şekil 1-8 EBOB in hesaplamasını yapan akış diyagramı. ....	4
Şekil 1-9 Faktöriyel işlemi yapan akış diyagramı.....	4
Şekil 1-10 Alternatif akım devresi. ....	5
Şekil 1-11 Yerçekimi ivmesinin yükseklikle değişimi probleminin akış diyagramı.....	6
Şekil 1-14 Bir bilgisayar programının kodlama aşamaları. ....	7
Şekil 4-1 for döngüsünün çalışması. ....	45
Şekil 4-3 while döngüsünün çalışması.....	49
Şekil 4-4 do ... while döngüsünün çalışması.....	50
Şekil 6-1 Dizi tanımı. ....	72
Şekil 7-1 Şirket personeline ait bilgiler. ....	83
Şekil 7-2 Bir nüfus cüzdanı örneğindeki bilgiler veri yapısının şeklini belirler. ....	84

C/C++ Ders Notları-A.Ü. Fizik Mühendisliği Bölümü

## ALGORİTMALAR DİZİNİ

Algoritma 1-1 Algoritma hazırlanışı. ....	2
Algoritma 1-2 İkinci dereceden polinomun köklerinin bulunması. ....	3
<b>Algoritma 1-3</b> İki sayının toplama işlemi. ....	3
Algoritma 1-8 EBOB. ....	4
Algoritma 1-9 Pozitif tamsayının faktöriyelinin bulunması. ....	4
<b>Algoritma 1-10</b> Seri AC devresi. ....	5
<b>Algoritma 1-11</b> Yerçekimi ivmesinin yükseklikle değişimi. ....	6
<b>Algoritma 2-1</b> Karakter ve tamsayı değişkenlerinin tanımlanması. ....	11
<b>Algoritma 2-2</b> Tek ve çift duyarlı değişkenler. ....	12
Algoritma 3-1 Bir değişkendeki değerin yazdırılması. ....	24
Algoritma 3-2 Derece cinsinden verilen açı değerinin radyan açı değerine dönüştürülmesi. ....	24
Algoritma 3-3 Fahrenheit sıcaklığının Celcius skalasına dönüştürülmesi. ....	25
Algoritma 3-4 Sayı sistemleri arasındaki dönüşüm. ....	26
Algoritma 3-5 Uydunun periyodu. ....	27
Algoritma 3-6 İki sayının çarpımı. ....	29
Algoritma 3-7 Dairenin alanının hesaplanması. ....	30
Algoritma 3-8 Küre hacminin hesaplanması. ....	30
<b>Algoritma 3-9</b> Açı tiplerinin belirlenmesi. ....	33
<b>Algoritma 3-10</b> Derslerin seçimi. ....	35
<b>Algoritma 3-11</b> Aynı ağırlığı kaldıran haltercilerden kazananın belirlenmesi. ....	37
<b>Algoritma 3-12</b> Kredi faiz hesabı. ....	39
Algoritma 3-13 Erkek ve kız öğrencilerin boy ortalamaları. ....	42
Algoritma 3-14 inch <sup>3</sup> lük hacim değerinin m <sup>3</sup> değerine dönüştürülmesi. ....	42
Algoritma 3-15 Günün saat dakika ve saniyeye dönüştürülmesi. ....	43
Algoritma 3-16 Dosyadan okunan bilgilerden maddelerin yoğunluğunun hesaplanması. ....	43
Algoritma 3-17 Dörtgen prizmanın köşegen uzunluğu. ....	44
<b>Algoritma 4-1</b> ( $i = 14i2$ ) ifadesini hesaplayan algoritm. ....	46
<b>Algoritma 4-2</b> $\sin x = x - x^3/3! + x^5/5! - x^7/7! + x^9/9! - \dots$ ifadesini hesaplayan algoritma. ....	47
Algoritma 4-3 Basket potasına değişik açılarda atışlar. ....	<b>Error! Bookmark not defined.</b>
<b>Algoritma 4-4</b> $\sin x = x - x^3/3! + x^5/5! - x^7/7! + x^9/9! - \dots$ ifadesini hesaplayan algoritma. ....	51
Algoritma 4-5 Bakteri kültürü nüfusunun sayımı. ....	51
Algoritma 4-6 Bakteri kültürü nüfusunun sayımı. ....	52
Algoritma 4-9 Rasgele sayı üretimi. ....	55
Algoritma 4-10 Değişken kuvvet için toplam işin hesaplanması. ....	57
Algoritma 5-5 Maddenin enerjiye çevrilmesi. ....	68
Algoritma 6-1 İki vektörün skaler çarpımı. ....	72
Algoritma 6-2 Zamana bağlı kuvvetin değişimi (Impuls). ....	74
Algoritma 6-3 Kütle merkezinin belirlenmesi. ....	76
Algoritma 6-4 Harmonik titreşkenin kinetik ve potansiyel enerjisinin zamanla değişimi. ....	77
Algoritma 6-6 Enlem ve boylamlara ait sıcaklık değerlerinin dosyadan okunması. ....	80

## PROGRAMLAR DİZİNİ

<b>Program 2-1</b> Karakter ve tamsayı değişkenlerin kullanılması.....	12
Program 2-2 Tek ve çift duyarlı sayılar. ....	12
<b>Program 2-3</b> Çizelge 2-8 deki işlemcilerin kullanımını gösteren C programı.....	15
Program 2-4 Hazır fonksiyonların kullanımı. ....	18
<b>Program 2-5</b> Karakter olarak tanımlanmış verinin tam sayıya dönüştürülmesi. ....	20
<b>Program 2-6</b> Tamsayı değeri karakter özellikli stringe dönüştürülmesi.....	20
<b>Program 2-7</b> Örnek bir C++ programı. ....	21
Program 3-1 Değişkendeki değerın ekrana yazdırılması. ....	24
Program 3-2 Derece-radyan açđ dönüşümü. ....	25
Program 3-3 Fahrenheit-Santigrat dönüşümü. ....	26
<b>Program 3-4</b> 10 tabanlı sayı sisteminden 16 tabanlı sayı sistemine geçiş. ....	26
<b>Program 3-5</b> Uydunun dünya çevresindeki dönme periyodunun hesaplanması.....	28
<b>Program 3-6</b> Biçimli giriş-çıkış. ....	29
<b>Program 3-7</b> Dairenin alanını hesaplanması. ....	30
<b>Program 3-8</b> Kürenin hacminin hesaplanması.....	31
<b>Program 3-9</b> C++ biçimli giriş-çıkış işlemi. ....	31
<b>Program 3-11</b> if ve getch() deyimlerinin kullanılışı. ....	32
<b>Program 3-12</b> go to deyiminin kullanılması. ....	34
<b>Program 3-13</b> switch () case yapısı ile ders tercihinin yapılması.....	35
<b>Program 3-14</b> $y(x) = \ln(1/(1-x))$ in değerlerinin if ve goto deyimleri ile hesaplanması. ....	36
<b>Program 3-15</b> if, switch-case yapısını kullanarak halterde 1nciyi belirleme.....	37
Program 3-16 Dosyaya veri yazan bir C++ programı. ....	39
Program 3-17 Dosyaya kredi faizlerini veri olarak yazan bir C++ programı. ....	39
Program 4-1 1 den 10 a kadar rakamların for döngüsü içinde yazdırılması. ....	46
Program 4-2 ( $i = 14i2$ ) ifadesinin hesaplanması.....	46
Program 4-3 $\sin x = x - x^3/3! + x^5/5! - x^7/7! + x^9/9! - \dots$ ifadesini hesaplayan C++ programı. ....	47
<b>Program 4-4</b> ( $i = 14i2$ ) ifadesini hesaplayan program. ....	48
<b>Program 4-5</b> Basket çemberinden topu geçirmek (for döngüsü ile)..... <b>Error! Bookmark not defined.</b>	
Program 4-9 Dosya sonuna ulaşıp ulaşılmadığını while ile kontrol eden bir C++ programı. ....	49
<b>Program 4-10</b> Bakteri kültürü nüfusunun sayımı. ....	52
Program 4-11 Bir dosyadan okunan verilerin işlendikten sonra başka bir dosyaya yazılması	53
<b>Program 4-12</b> C++ da dosyadan veri okuma. ....	54
Program 4-13 Ortalama ivme hesabı.....	54
Program 4-18 Rasgele sayıların türetilmesi. ....	56
Program 4-19 Değişken kuvvet için toplam işin hesaplanması (C). ....	57
Program 4-20 Değişken kuvvet için toplam işin hesaplanması. ....	57



Program 4-21 Bazı besinlerin kalori, protein, yağ, VitA ve kalsiyum değerlerinin dosyadan okunması. ....	59
Program 5-1 Sıcaklık değerlerinin Fahrenheit tan Celcius a fonksiyon ile çevrilmesi. ....	62
<b>Program 5-2</b> Reaktör kontrolü programının fonksiyon kullanılarak yazılması. ....	63
<b>Program 5-3</b> Fonksiyonların kullanımı ve not hesaplanması. ....	64
Program 5-4 Fonksiyon kullanarak isimlerin ekrana yazdırılması. ....	65
Program 5-5 Bir malın KDV hariç fiyatı. ....	66
Program 5-6 Klavyeden girilen iki sayıdan büyüğünün ekrana (fonksiyon kullanılarak) yazdırılması. ....	67
Program 5-12 Maddenin enerjiye dönüşümü. ....	68
Program 5-13 gotoxy fonksiyonunun kullanılması. ....	68
<b>Program 6-1</b> İki vektörün skaler çarpımı. ....	72
<b>Program 6-2</b> Karakterlerin dizilere aktarılması. ....	73
<b>Program 6-3</b> Karakterlerin dizi elemanlarına aktarılması. ....	73
<b>Program 6-4</b> İmpuls denklemi kullanılarak son hızın hesaplanması. ....	74
Program 6-5 Kütle merkezinin hesaplanması. ....	76
Program 6-6 Harmonik titreşkenin kinetik ve potansiyel enerjisinin zamanla değişimi. ....	78
Program 6-8 Enlem-boylam sıcaklıklarının dosyadan okunup diziye aktarılması. ....	80
Program 7-1 Filmlerin adının ve yapım yılının veri yapısı ile saklanması. ....	85
Program 7-2 Arabalara ait bilgilerin veri tabanı. ....	86
Program 7-3 Hava durumu bilgileri (veritabanı). ....	87
Program 7-4 Tarihleri veri tabanı yapısı kullanarak saklamak için hazırlanması düşünülen bir C programı aşağıda verilmektedir. ....	88
Program 7-5 Bağışçılarının bilgilerinin saklanması. ....	88
Program 7-7 Kişi bilgilerinin veri yapısı ile saklanması. ....	90
Program 7-9 Dörtgen alanının ve çevresinin sınıf yapısı kullanılarak hesaplanması. ....	92
Program 7-10 x-ekseni boyunca zamana bağlı hareket : $x(t)=0.20t-5.0t^2+7.5t^3$ ....	93
Program 7-15 Bir dairenin alanı ve çevresinin class kullanılarak hesaplanması. ....	94

## ÇİZELGELER DİZİNİ

Çizelge 1-3 Bir bilgisayar programının kodlama aşamaları.....	6
Çizelge 2-1 C++ da kullanılan veri tipleri.....	9
Çizelge 2-2 C++ dilinde değişmez ve değişkenlerde kullanılacak karakterler. ....	10
Çizelge 2-3 C++ programlama dilinde kullanılan aritmetik işlemciler. ....	13
Çizelge 2-4 C++ programlama dilinde aritmetik işlemlerde öncelik sırası. ....	13
<b>Çizelge 2-5</b> C++ programlama dilinde kullanılan atama sembolleri. ....	14
<b>Çizelge 2-6</b> C++ programlama dilinde önce işlem sonra artırma veya önce artırma sonra işlem. ....	14
<b>Çizelge 2-7</b> C++ programlama dilinde artırma veya eksiltme işlemleri ( $a=4$ , $b=5$ ).....	15
Çizelge 2-8 C/C++ da önce değer değiştirme sonra işlem veya önce işlem sonra değer artırma. ....	15
<b>Çizelge 2-9</b> C++ programlama dilinde kullanılan karşılaştırma sembolleri. ....	17
Çizelge 2-10 C++ da kullanılacak bazı matematiksel fonksiyonlar (<math.h>).....	18
Çizelge 3-1 C/C++ ta printf ve cout ile kullanılan escape karakter komutları. ....	23
Çizelge 3-2 printf komutunda yazım biçiminin ayarlanması.....	23
Çizelge 3-4 Sistem başlangıç dosyaları.....	31
Çizelge 3-5 Halter de 1nciyi belirleme kriteri.....	37
Çizelge 3-6 C++ programlama dilinde rezerve edilmiş fonksiyon isimleri (Keywords).....	41
Çizelge 4-1 Zamana bağlı olarak sürat değerleri. ....	55
Çizelge 4-2 x-ekseni boyunca hareket eden cisme etkiyen değişken kuvvet.....	56
Çizelge 4-3 Çeşitli besinlerin kalori, protein, yağ, VitA ve kalsiyum değerleri. ....	58
Çizelge 5-1 Fonksiyon alt programlarının hazırlanması için algoritma.....	62
Çizelge 6-1 Zaman bağlı olarak kuvvetin değişimi. ....	74
Çizelge 6-2 Değişik konum ve kütle değerlerine sahip cisimlerin kütle merkezinin hesabı. ....	75
Çizelge 6-3 Batı enlemi ile kuzey boylamlarındaki meteorolojik veriler. ....	80

# 1. BÖLÜM (1nci hafta)

## 1.1 YAZILIM MÜHENDİSLİĞİ ve C++

Problemleri çözümlmek için geliştirilen programlarla ilgili işlemler karmaşıktır. Bilgisayar programcılığında hayal gücü, bilimsel çözüm tekniklerinin ve yöntemlerinin uygulanması (matematiksel formüllerin kullanılması ve bunların işlenmesi) yaratıcılık ve ustalık için gereklidir. **Yazılım mühendisi**, problemleri bilgisayar programı haline getirirken sanatsal ve bilimsel teknikleri kullanan kişidir. C++ programlama dili bu tür özellikleri kolayca kullanabilmek için geliştirilmiş yüksek seviyeli bir bilgisayar programlama dilidir.

Günümüz yazılımcısı, çoklu işlem yapabilen, çok kullanıcıya hizmet verebilen, hızlı, bellek tasarrufu sağlayan, değişik bilgi depolama yöntemlerini sunan, esnekliği olan, modüler programların eklenebileceği, operatörlerin kolayca kullanılabileceği, güzel görüntü gibi amaçlarla; makine dili ya da sistem düzeyinde araştırma yapmakta ve bilgisayarların tüm yeteneklerinden yararlanmak istemektedir. Yüksek seviyeli programlama dilleri bu tür istekleri tam anlamıyla karşılayabilecek düzeyde değildir.

1969 yılında, **Ken Thompson** UNIX işletim sistemini AT&T nin Bell Laboratuvarlarında makinalara inşa ederken PDB-7 makina dilini kullanmaktaydı. UNIX işletim sistemi yazılımını daha üstün özelliklere sahip DEC (Digital Equipment Corporation) PDP-11 makinesine taşımakta zorlandı. Binlerce satırlık makina dilinde yazılmış kodları bu yeni makineye aktarmakta güçlükler ortaya çıkıyordu. Oysa o, yazdığı işletim sisteminin değişik makinalarda çalışmasını istiyordu. Bu yüzden UNIX işletim sistemini yüksek seviyeli bir dille yazmaya başladı. İşletim sistemini yazarken kullandığı programlama dilinin adına B dedi. B dilinde yazılmış olan UNIX işletim sistemini başka makinalara aktarmadaki zorluklar devam edince, Bell Laboratuvarlarında çalışan **Dennis Ritchie** 1972 yılında, B dilinin %90 nını değiştirerek yeni bir programlama dili geliştirdi ve adına C dedi. C derleyicisi çok ucuz ve kullanımı çok kolay olduğu için, kısa sürede üniversitelerde, yüksek okullarda ve daha sonra iş çevrelerinde kullanılan popüler bir programlama dili oldu. Bell Laboratuvarlarında çalışan diğer bir araştırmacı **Bjarne Stroustrup**, C nin geliştirilmesi projesi dahilinde bu dildeki bazı zorlukları ortadan kaldırıp, nesneye yönelik programlama dili haline getirdi. 1983 yılında, yeni yüzlü bu programlama diline **C with Classes** adı verildi. Daha sonraları C++ olarak yeniden adlandırıldı ve günümüzde en popüler bilgisayar programlama dillerinden biri haline geldi.

C++ programlama dilini kullanarak iş, finans, posta listeleri, faturalama, ödeme, hesap takibi, rezervasyon sistemleri (havayolları, otobüs, sinema, araba kiralama, vb.), kelime işlemciler, veri yönetimi, elektronik fon transferi, otomatik para makinaları, ev bankacılığı, elektronik posta, finansal planlama, sigorta işlemleri, endüstride robot yöntemi, CAD, CAM, CIM, savunma sistemleri, uzay programları, hava tahminleri, otomatik trafik kontrol sistemleri vs birçok alanda program yazmak mümkündür.

C++ Programlama dili, blok yapılı ya da yapısal (hierarchy) bir programlama dilidir. Burada kullanılan blok, belirli bir işlev için bir araya getirilmiş bir veya birden fazla komut cümlesinin oluşturduğu program parçası için kullanılmaktadır. Bu özellik programcının program üzerindeki hakimiyetini artıran, karışıklığı engelleyen bir yapıdır.

Yapısal programlama dillerinin en büyük özelliği oldukça düzenli ve okunabilir olmalarıdır. Yapısal dil ile yazılmış bir program incelendiğinde, programın ana bileşenleri rahatça görülebilir ve birbirlerinden kolayca ayırt edilebilir durumdadır. Program üzerindeki yeni düzenlemelerin çok kolay yapılabilmesi, bu özelliğin getirdiği önemli bir avantajdır. Yapısal programlama dillerine örnek olarak Pascal, Ada, C, Modula-2 gibi programlar verilebilir. Yapısal olmayan programlama dillerine ise Fortran, Cobol ve Basic gibi programlar örnek verilebilir.

## 1.2 ALGORİTMA ve AKIŞ DİYAGRAMLARI

Algoritma veya akış diyagramları bir problemin satırlar veya şekillerle nasıl çözüleceğini gösterir. Algoritmalar veya akış diyagramları hazırlayarak problemlerin çözümleri daha kolay hale getirilebilir ve bilgisayar programı haline kolayca dönüştürülebilir. Bir bilgisayar programı, algoritma ve verilerin birleşimidir: **Program=algoritma+veri.**

Günlük hayatımıza başladığımızda kafamızda o günün, ayın veya yılın bir planı veya programı varsa bu planı uygulamaya çalışıyorsak bilgisayarları çalıştırmak için programların planlanması ve yazılması gerekmektedir. Program yazmak için birçok yöntem kullanılabilir veya yol izlenebilir. Aşağıda, bu işin başlangıcında olanlar için nasıl bir yol izleneceği (algoritma) kısaca verilmektedir. Algoritma hazırlama konusunda literatürde bir çok kaynak bulunabilir.

**Örnek 1-1** Bir problemin algoritmasını hazırlamak için aşağıda işlem sırası verilmektedir.

**Algoritma 1-1** Algoritma hazırlanışı.

1. Problemin anlaşılması,
2. Problemin matematiksel modelinin oluşturulması,
3. Giriş bilgilerinin değerlendirilmesi,
4. Giriş verileri için olasılıkları analiz etmek,
5. Giriş verilerini algoritma üzerinde test etmek,
6. Olasılıkları algoritma üzerinde test etmek,
7. Uygun bir programlama dili seçip programı yazmak ve çalıştırmak,
8. Sonuçları elde etmek,
9. Sonuçları benzer programların sonuçları ile karşılaştırmak,
10. Programı kullanıma sunmak.

Yukarıdaki algoritmaya satırlar eklenebilir, çıkarabilir, sorgulamalar, tekrarlamalar, satır atlamaları yapılabilir. Bazı satırlardaki işlemler uzun sürebilir. Bu durumda işlem kontrollerinin yapılması gerekebilir. Bu işlemler sırası problemin yapısına uygun bir şekilde değiştirilebilir.

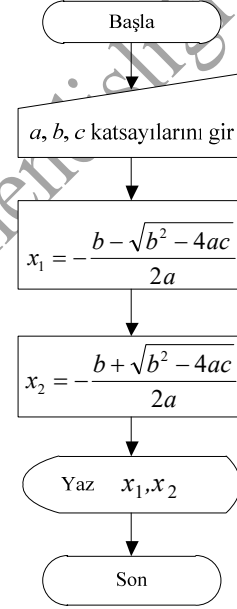
Algoritmalar bir problemin bilgisayar programı haline getirilmesi aşamasında çok önemlidir. Programlama konusunda ilerleyebilmek için problemlerin çözümlerinin algoritmalar üzerinden yapılması yararlı olacaktır. Bu kesimde bazı problemlerin aloritmaları ve/veya akış diyagramları verilerek problemlerin kolayca anlaşılması sağlanacaktır.

**Örnek 1-2**  $ax^2+bx+c=0$  şeklindeki ikinci dereceden bir polinomun köklerini hesaplayan bir algoritma (işlemler dizisi) ve akış diyagramı aşağıdaki gibi verilebilir.

**Algoritma 1-2** İkinci dereceden polinomun köklerinin bulunması.

1. Başla,
2. a, b, c katsayılarını gir,
3.  $x_1 = (-b - \sqrt{b^2 - 4ac}) / (2a)$
4.  $x_2 = (-b + \sqrt{b^2 - 4ac}) / (2a)$
5.  $x_1$  ve  $x_2$  köklerini yazınız
6. Son

Algoritma başlama yeri (1. Başla,) belirtilir. Adımlar yani yapılacak işlem sırası artan rakamlara göre belirlenir. Giriş verileri matematiksel model veya formüllerde kullanılarak hesaplanan değerler a, b, c,  $x_1$  ve  $x_2$  sembollerinde (2, 3 ve 4ncü adımlar) saklanır.  $x_1$  ve  $x_2$  değişkenlerinde saklanan değerler ekran, kağıt vs ortamlarına aktarılır (5nci adım) ve algoritma sonlanır. Aynı problemin çözümü Şekil 1-1 de akış diyagramı olarak verilmektedir. Burada işlemlerin sırası ve yönü oklarla, işlemler dörtgen, paralelkenar gibi şekille gösterilmiştir.



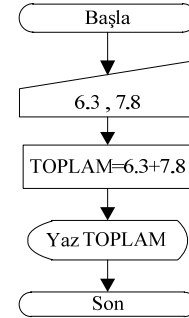
**Şekil 1-1** İkinci dereceden bir polinomun köklerinin hesaplanmasının akış diyagramı.

**Örnek 1-3** İki sayının toplamını yapan algoritma ve akış diyagramı aşağıda ve yanda verilmektedir. Akış diyagramında 6.3 ve 7.8 rakamlarının toplama işlemi yapılmakta ve toplama sonucu TOPLAM değişkenine aktarılmaktadır. Sonraki aşamada ise elde edilen sonuç ekrana yazdırılıp, işlemler sonlanmaktadır.

**Şekil 1-2** Örnek 2-1 için iki sayının toplamını yapan akış diyagramı.

**Algoritma 1-3** İki sayının toplama işlemi.

1. Başla İkiSayıToplamı
2. 6.3 ve 7.8 rakamlarını Gir
3. TOPLAM=6.3+7.8
4. Yaz TOPLAM
5. Son

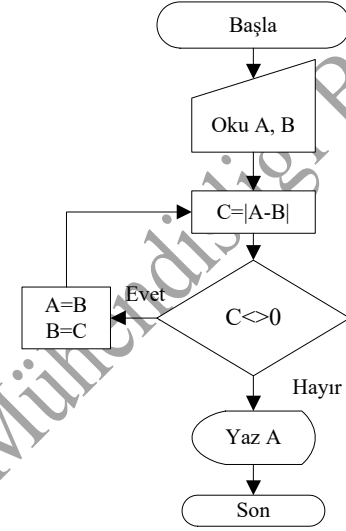


Yapılacak işlemlere yani algoritmaya başlama noktası (1. Başla İkiSayıToplamı,) belirtilerek girilir. Adımlar artan rakamlara göre sıralanmıştır. Veriler (6.3 ve 7.8) bilgisayarın belleğine aktarılır (2nci adım), matematiksel model veya formül (3ncü adım) ile toplama işlemi yapılır ve sonuç TOPLAM değişkenine aktarılır. 4ncü adımda semboldeki değer çıkış ortamına aktarılır ve işlemler dizisi 5nci adımla sonlanır. Aynı işlemler

#### Algoritma 1-4 EBOB.

1. Başla
2. A, B değerlerini oku
3.  $C=|A-B|$
4. Eğer  $C=0$  ise Git 8
5.  $A=B$
6.  $B=C$
7. Git 3
8. Yaz A
10. Son

Şekil 1-3 EBOB in hesaplamasını yapan akış diyagramı.



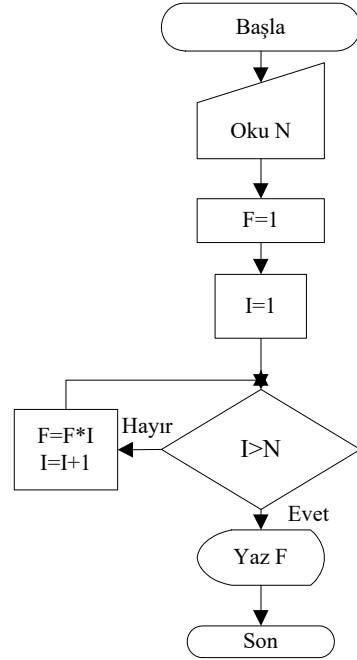
Örnek 1-4 Aşağıdaki algoritmayı ya da akış diyagramını  $N=4$  değeri için izlerseniz 4 ün faktöriyel ( $4!=1 \times 2 \times 3 \times 4$ ) olan 24 ü hesaplanabilir. Algoritmayı ve akış diyagramını girilen sayının pozitif olup olmadığını kontrol ettikten devam edecek şekilde yeniden düzenleyebilirsiniz. Bunun için sayı girildikten hemen sonra girilen sayının sıfıra eşit olup olmadığı, sıfırdan küçük olup olmadığı sorgulanabilir.

#### Algoritma 1-5 Pozitif tamsayının faktöriyelinin bulunması.

1. Başla
2. Oku N
3.  $F=1$
4.  $I=1$
5. Eğer  $I > N$  ise Git 9
6.  $F=F \times I$
7.  $I=I+1$
8. Git 5
9. Yaz F
10. Son

Şekil 1-4 Faktöriyel işlemi yapan akış diyagramı.

N değişkenindeki değer sıfırdan büyük ise 5 ile 8nci satırlar arasında çarpma işlemi ( $F=F \times I$ ) yapılarak faktöriyel hesaplanır. Çarpma işleminden sonra sayaç değişkeni 1 artırılır. 5nci satırdaki koşul sağlanana kadar (yani  $I > N$  olana kadar) bu tekrarlamaya işlemine devam edilir. Burada işlem önceki değer



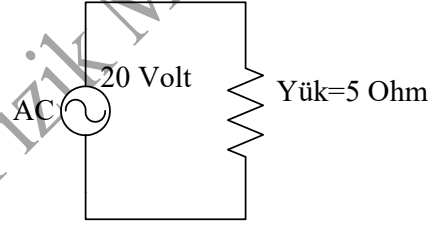
kendisi ile çarpımıdır (yani  $F \cdot I$ ). Faktöriyel hesabı çarpma işlemi içerdiği için 3ncü satırda  $F$  değişkenine öncelikle 1 değeri aktarılır.  $I$  sayacı ile işlem sayısı kontrol edilerek hesaplanan sonuç 9ncü satırda yazdırılır ve program sonlanır. Ayrıca Şekil 1-4 deki akış diyagramı takip edilerek işlemin nasıl gerçekleştirildiği görülebilir.

**Örnek 1-5** Seri bağlı sinüzoidal güç kaynağı ( $V$ ) ve yükten ( $Z$ ) oluşan bir alternatif akım (AC) elektrik devresinde çeşitli karakteristik değerleri hesaplayan bir algoritma aşağıda verilmektedir ( $\theta = \text{teta} = 30^\circ$ ):

$I = V/Z$  (Amper)-devreden geçen akım  
 $P = V I \cos(\text{teta})$  (Watt)- devreye verilen gerçek güç  
 $Q = V I \sin(\text{teta})$ -(Watt) reaktif güç  
 $S = V I$ -(VA)-görünen güç  
 $PF = \cos(\text{teta})$ -yükün güç faktörü

**Algoritma 1-6** Seri AC devresi.

1. Başla
2.  $V=20.0$
3.  $Z=5.0$
4.  $\text{teta}=30.0$
5.  $I=V/Z$
6.  $P=V \cdot I \cdot \cos(\text{teta})$
7.  $Q=V \cdot I \cdot \sin(\text{teta})$
8.  $S=V \cdot I$
9.  $PF=\cos(\text{teta})$
10. Yaz  $V, Z, \text{teta}, I, P, S, PF$
11. Son



**Şekil 1-5** Alternatif akım devresi.

Algoritmanın 1nci satırıyla programın başlama yeri belirlenmiştir. Daha sonraki 2, 3 ve 4ncü satırlarda değişkenlere ilk değerleri atanmıştır. 5, 6, 7, 8 ve 9ncü satırlarda bazı matematiksel işlemler gerçekleştirilmektedir (matematiksel modeller). 10ncü satırda elde edilen değerler ekrana yazdırılıp iş akışı tamamlanmaktadır. Bu aşamadan sonra algoritmanızı elle kontrol ederek satır satır işlemleri yapabilirsiniz. Elde ettiğiniz sonuçlar doğru ise bir programlama dili seçerek (Fortran, Delphi, Visual Basic vs) algoritmanızı bu programlama diline çevirebilirsiniz. Fortran bilgisayar programı yazılmışsa çalışma aşamasına kadar çeşitli aşamalardan geçilir.

**Örnek 1-6** Dünyanın çekim kuvvetinden dolayı oluşan kütleli yerçekimi ivmesi yeryüzeyinden yükseldikçe  $g = -G \cdot M / (R+h)^2$  denklemindeki gibi değişmektedir:Denklemindeki

$G=6.672 \times 10^{-11} \text{ Nm}^2/\text{kg}^2$  (yerkürenin çekim sabiti),  
 $M=5.98 \times 10^{24} \text{ kg}$  (yer kürenin kütlesi),  
 $R=6371 \text{ km}$  (yerkürenin yarıçapı),  
 $h$ -metre cinsinden yeryüzeyinden yükseklik.

Denklemdede kullanılacak deęerlerin aynı birimde olmasına dikkat edilmiş ve metre biriminde verilmiştir. Bunun için kilometre biriminde verilen deęerler metre birimine dönüştürülür. Yeryüzeyinden başlayarak her 400km yükseklikte yerçekimi ivmesinin deęerini 40000km ye kadar hesaplayıp yazacak bir algoritma ve akış diyagramı aşağıda verilmektedir.

**Algoritma 1-7** Yerçekimi ivmesinin yükseklikle deęişimi.

1. Başla
2.  $G=6.672 \times 10^{-11}$
3.  $M=5.98 \times 10^{24}$
4.  $R=6371 \times 10^3$ ,
5.  $h=0$
6. Eğer  $h > 40000 \times 10^3$  Git 11
7.  $g = -G * M / (R+h)^2$
8. Yaz  $h, g$
9.  $h = h + 400 \times 10^3$
10. Git 6
11. Son

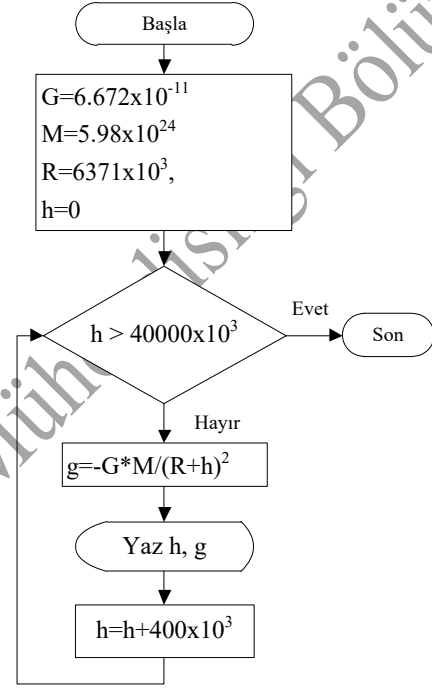
**Şekil 1-6** Yerçekimi ivmesinin yükseklikle deęişimi probleminin akış diyagramı.

Yukarıdaki algoritmada sabit deęerler 2, 3, 4 ve 5 nci satırlarda deęişmezlere ilk deęerleri atanır. 6ncı satırda ise belirtilen yükseklik miktarlarında hesaplanacak olan matematiksel formül belirtilmektedir. 7nci satırda istenilen yükseklik deęerine ulaşıncaya algoritmanın devam edip etmeyeceğine karar verilir. Maksimum yüksekliğe ulaşılmayınca önce yerçekimi ivmesi  $g$  hesaplanır ve bu deęer ekrana yazdırılır. Denklemdaki - (eksi) işareti yerçekimi ivmesinin aşağıya doğru olduğunu belirtir. Daha sonra 6nci satıra dönülerek yükseklik deęeri  $h$  istenilen miktar kadar artırılır ve 7nci satırda maksimum deęerine ulaşıp ulaşılmadığı kontrol edilir. Algoritma, maksimum yüksekliğe ulaşıncaya 11nci satıra satarak durur. Akış diyagramından problemin çözüm yolu gösterilmektedir. Yerçekimi ivmesinin yönü yere doğru olduğu için - (negatif) işaretlidir.

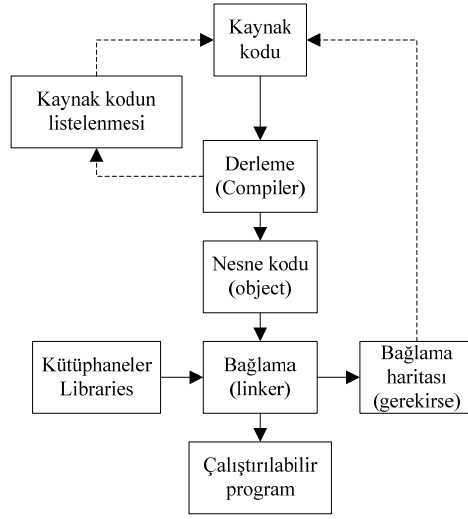
Algoritma veya akış diyagramları hazırlanıp test edildikten sonra bilgisayar programı yazım aşamasına geçilebilir (buna kodlama denir). Bilgisayar programının yazım esnasında ve sonrasında aşağıdaki Çizelge 1-1 deki veya Şekil 1-14 deki aşamalardan geçilir:

**Çizelge 1-1** Bir bilgisayar programının kodlama aşamaları.

Kaynak program → Derleyici → Nesne program → Bağlayıcı → Program kodu  
(yüksek seviyeli dil) (yazım kuralları) (makina dili) Çalışabilir hale getirmek







**Şekil 1-7** Bir bilgisayar programının kodlama aşamaları.

Kaynak kod (algoritma) yani bilgisayar programı bir metin editörü aracılığı ile yazılır. Günümüz derleyicilerin çoğu bir metin editörü içermekte ve yazım anında komutlar ve diğer bileşenler kolayca kontrol edilmektedir. Kodun yazım aşamasında hataları varsa düzeltilir. Kaynak kod hazırlandıktan sonra derleyicinin (compiler) oluşturduğu nesne dosyası (object file) bir bağlayıcı (linker) aracılığı ile bilgisayarın çalıştırabileceği (execute) seviyeye getirilir. Nesne dosyası bağlayıcı programlar aracılığı ile makine diline (yani bilgisayarın anlayacağı hale) çevrilip çalışmaya hazır hale getirilir.

### 1.3 SORULAR

1. Bir sınıftaki 20 öğrencinin 10 tanesi erkek, 10 tanesi de kızdır. Kızların boy ortalaması 135 cm iken erkeklerin boy ortalaması 130 cm dir. Sınıfın boy ortalamasını hesaplayan bir algoritma yazınız. Burada öğrenci sayısı belli olduğu için başlangıcı ve bitiş değerleri belirli döngü kullanılmalıdır.
2. Bir malın KDV dahil fiyatı 500 TL ise, malın vergisiz (KDV) değerini hesaplayan bir algoritma ve akış diyagramı hazırlayınız (KDV=%17 ise  $KDVSIZ = KDV / (1 + KDV/100)$ ).
3. Bir kablo döşeme firması için çalışan işçi, yaptığı her döşeme işi başına 10 TL ve ayrıca metre başına 1.5 TL lik para almaktadır. Firma yöneticisi bu işçinin bir ay içinde yapmış olduğu döşeme işinin sayısını ve toplam kablo uzunluğunu metre olarak girip işçiye ödenecek parayı hesaplamak istemektedir. Bunun için bir algoritma geliştiriniz.

4. Klavyeden girilen  $n$  değerine kadar  $\sum_{i=1}^n \frac{1}{i}$  şeklinde toplam hesabı yapan bir algoritma yazınız.

