

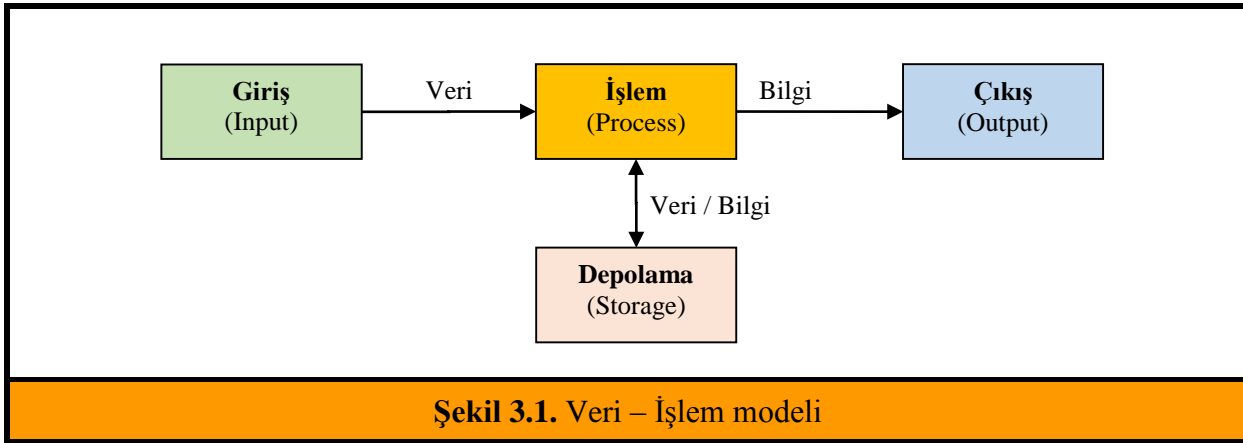


3. Bölüm

Algoritmalar

3.1. Veri ve Bilgi

Şekil 3.1’de bilgisayar sistemin temelini oluşturan veri işlem modeli görülmektedir. Hesaplama, saklama gibi çeşitli işlemler amacıyla bilgisayara verilen sayı, yazı, resim, ses, ölçüm vb. değerlerden oluşan her türlü sayısal, alfasayısal bilgiler veri olarak adlandırılmaktadır. Verinin bilgisayar tarafından işlenmesiyle enformasyon/bilgi elde edilmektedir. Ayrıca bir işlem sonucu elde edilen enformasyon başka bir işlem için veri olarak kullanılabilir.



Şekil 3.1. Veri – İşlem modeli

3.2. Veri Yapıları (Data Structures)

Veri Yapısı

Bellek byte adı verilen sıra halinde dizilmiş hücrelerden oluşmaktadır. Bilgisayar ortamında veriler üzerinde işlem yapabilmek için öncelikle onların bellek üzerine alınması gerekir. Bu amaçla verilerin belleğe belirli kurallara uyarak alınmasını ve belirli şekillerde düzenlenmesini sağlayan temel yapıtaşları veri yapıları olarak adlandırılmaktadır.

Kısaca veri yapısı, birincil (ana bellek ...) veya ikincil (disk ...) bellek üzerinde verilerin tutulma şeklini göstererek onları anlamlı kılan yapılardır. Temel olarak kullanılan tamsayı, kesirli sayı, karakter, karakter dizisi, dizi vb. olmak üzere basit ve karmaşık çeşitli veri yapıları mevcuttur. Ayrıca isteğe bağlı olarak programlama dillerinin verdiği imkânlar çerçevesinde yeni veri yapılarının da oluşturulması mümkün olabilmektedir.

Veri yapıları, bilgisayar belleğinin verimli ve etkin bir şekilde kullanılması için gereklidir. Veri yapılarının iyi düzenlendiği bilgisayar programları hem daha hızlı çalışmakta hem de bilgisayar belleğini daha verimli kullanmaktadır.

Bellekte bulunan bir veri, veri yapısına bağlı olarak farklı farklı anlamlar kazanabilmektedir. Tablo 3.1’de ‘0100 0001’ verisi için veri yapısına bağlı olarak ifade ettiği veriler görülmektedir.

Tablo 3.1. Veri yapısına bağlı olarak verinin anlamı

Sembolik Bellek Verisi	Veri Anlamı	
0100 0001	Onlu (Decimal)	65
	İkili (Binary)	0100 0001
	Sekizli (Octal)	11
	Onaltılı (Hexadecimal)	41
	ASCII	A
	BCD (Binary Code for Decimal)	41

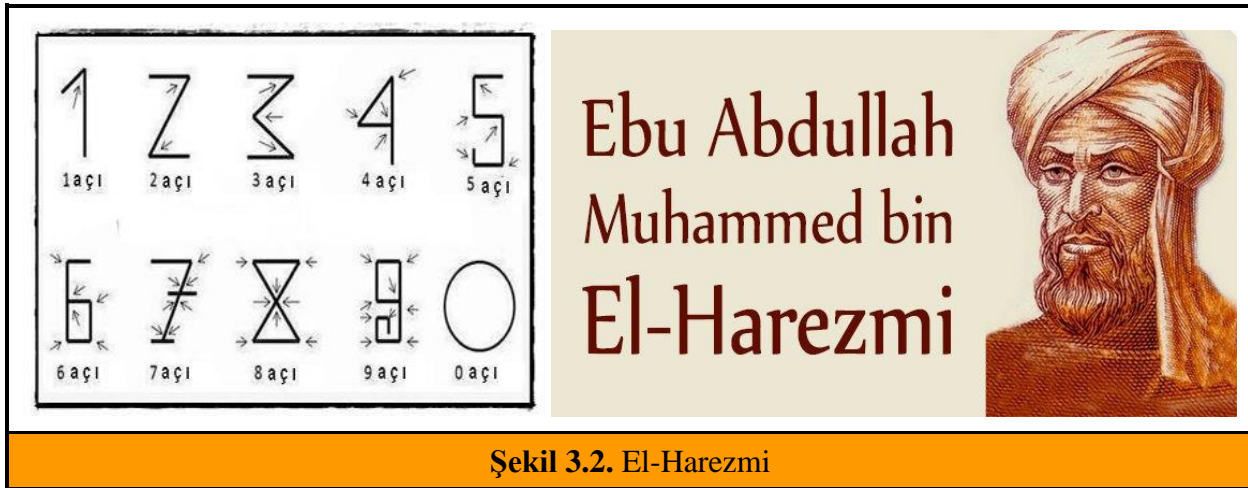
Veri Modeli (Data Model)

Verilerin ilişkisel, hiyerarşik, sırasal, ağ gibi çeşitli şekillerde düzenlenmesi amacıyla kullanılan yapı, kavram ve işlemler bütününe veri modeli denir.

3.3. Algoritma Temelleri

Algoritmanın Tarihçesi

Algoritma ve Cebir kavramlarının babası 9. yüzyıl başlarında (825) yaşayan ve matematik, gökbilim ve coğrafya alanlarında çalışmalar yapmış olan bilim adamı **Ebu Abdullah Muhammed bin Musa el-Harezmi**'dir (Şekil 3.2).



Matematik biliminde büyük katkılar sağlayan **El-Harezmi**'nin cebir alanında yazmış olduğu "Hisab el-cebir ve el-mukabala" adlı kitabı, algoritmik yaklaşımlara dayalı dünyanın ilk cebir kitabı olarak bilinir.

Algoritma

Algoritma, bilgisayar ve matematik biliminde, bir problemin çözümüne yönelik olarak geliştirilmiş, belli bir başlangıç ve bitiş noktası bulunan adımlardan oluşmaktadır. Algoritmalar hem basit yapıda olmalı hem de en az sayıda adım ile çözüme ulaştırabilmelidir.

Algoritmalar genel olarak şu bileşenlerden oluşur:

- Problem tanımı
- Başlangıç Şartlarının tanımı
- Parametre tanımı
- Çalıştırma
- Sonuçlar

Algoritma Analizi

Bir problemin çözümüne ilişkin olarak birden çok algoritma çözümü olabilir. Algoritma analizi, bir problemin çözümüne yönelik olarak geliştirilmiş algoritmalar arasından en uygun olanın araştırılarak seçilmesi işlemidir.

En Kötü Durum (Worst Case) Zamanı

Verilerin diziliminin en kötü olduğu durumda, mevcut algoritma en uzun zamanda çözüm üretir. En Kötü Durum Zamanı, en kötü veri diziliminde, algoritmanın çözüm için harcayacağı zamandır.

Karmaşıklık Analizi

Algoritma karmaşıklığı, algoritmanın en kötü durum zamanında çözümü elde edebilmek için gerekli maksimum adım sayısıdır. Giriş verilerinin dizilimine bağlı olarak maksimum adım sayısı elde edilebilmektedir.

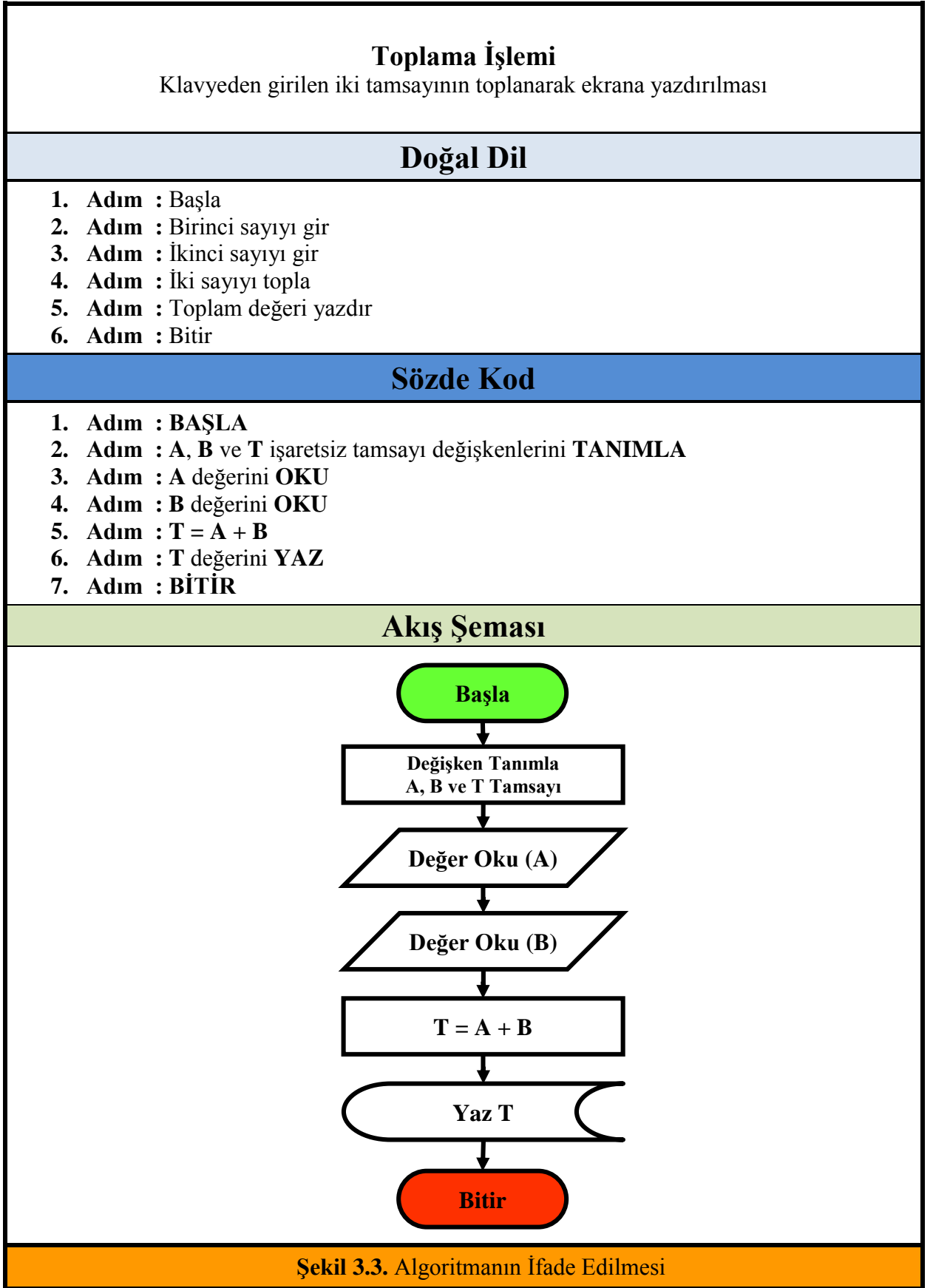
Bir problemin çözümüne yönelik olarak mevcut algoritmaların analizlerinin yapılması gerekir. Algoritmaların performanslarının belirlenerek karşılaştırılmasında karmaşıklık analizi kullanılır.

3.4. Algoritma İfade Şekilleri

Algoritmalar üç farklı şekilde ifade edilebilmektedir:

- a. Doğal Yazı Dili : Normal konuşma dili kullanılarak ifade etme şeklidir.
- b. Söзде Kod (Pseudo-Code) : Programlama dilinin detaylarından uzak bir şekilde, konuşma dilinin belirli kurallar çerçevesinde ve adımlar halinde kısa bir şekilde ifade edildiği komutlar kümesidir.
- c. Akış Şeması : Şekil ve semboller yardımıyla ifade etme şeklidir.



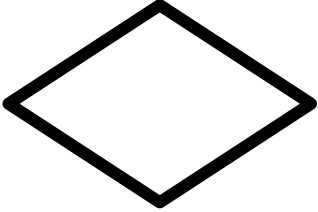

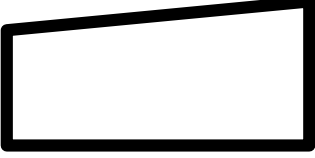




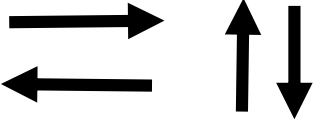


Şekil 3.3'te 2 sayının toplanmasına yönelik geliştirilen algoritmanın farklı şekillerde ifade edilmesi görülmektedir.



3.5. Akış Şemaları

Akış Şeması

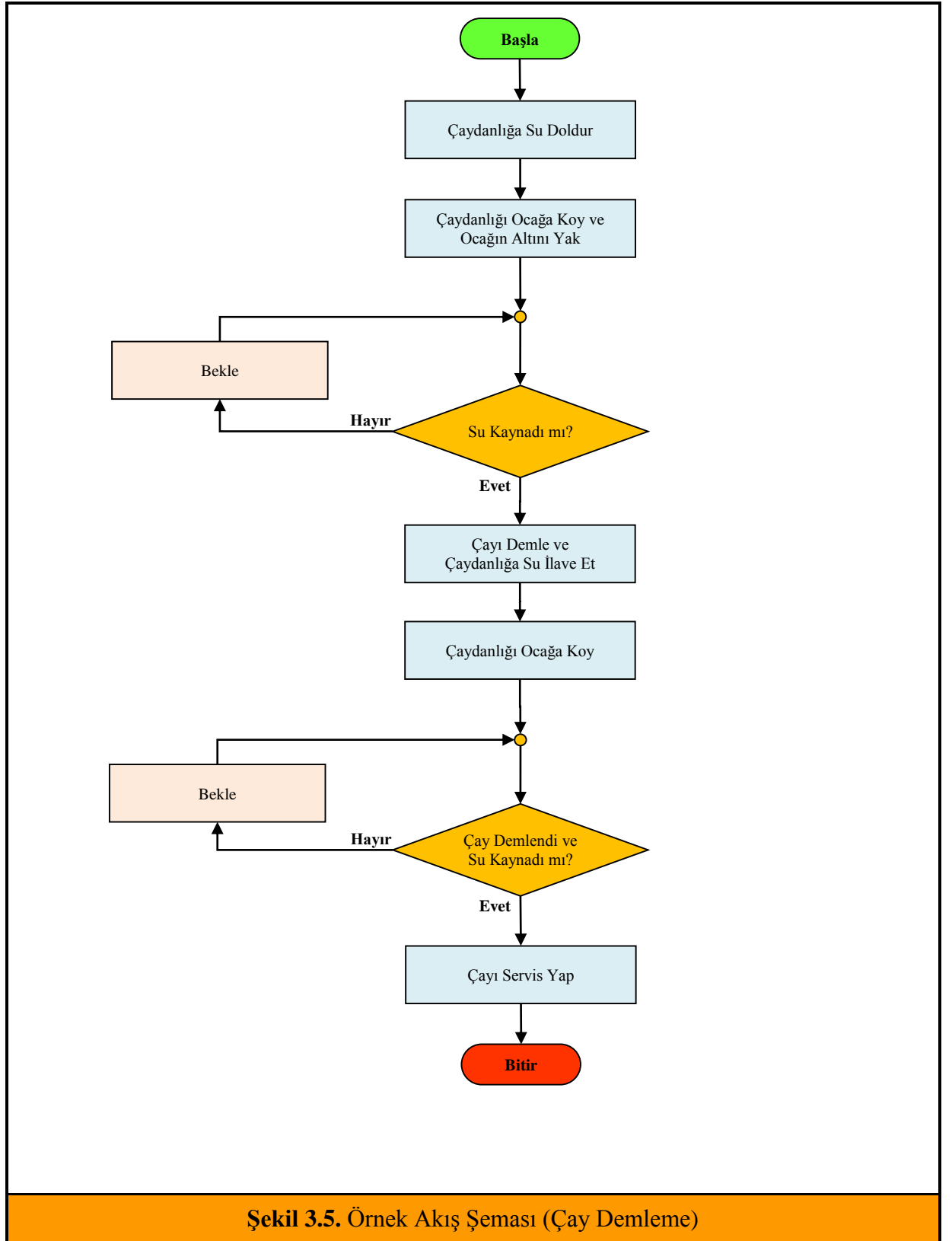
Akış şeması, algoritmanın geometrik semboller yardımıyla ifade edilmesidir. Akış şemalarının hazırlanmasında sıkça kullanılan semboller Şekil 3.4'te görülmektedir.

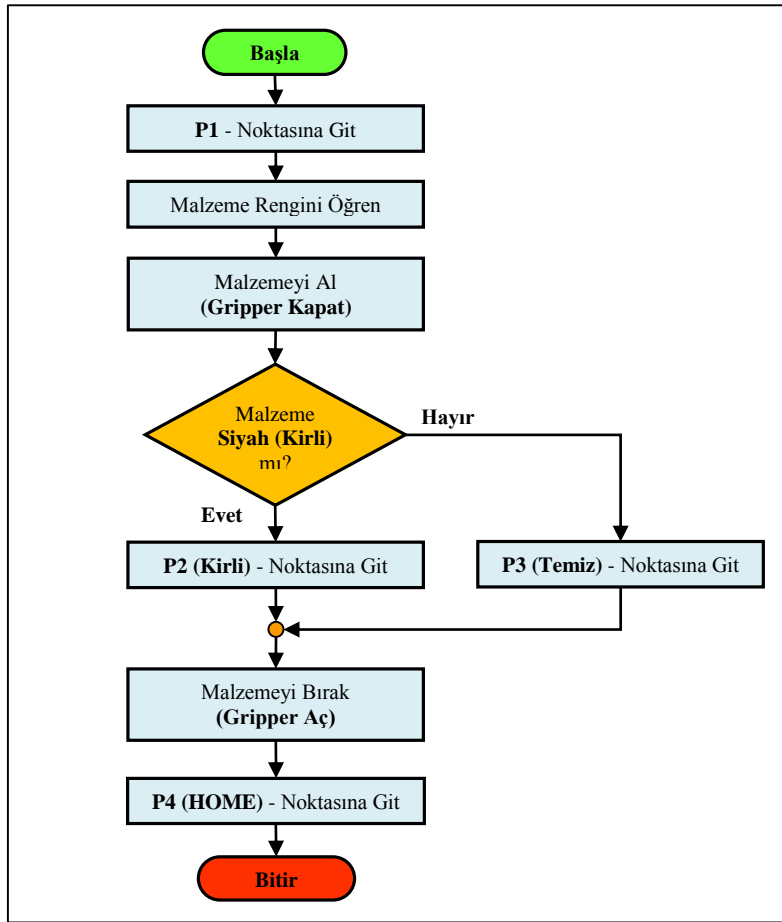
		
Başla / Bitir	İşlem	Karar
		
Genel Giriş / Çıkış	El ile Giriş (Klavye)	Çevrim / Döngü
		
Yazıcı Çıktısı	Ekran Çıktısı	Altprogram / Yordam
		
Akış Yönü	Bağlayıcı	Sayfa Dışı Bağlantı

Şekil 3.4. Akış Şeması Sembolleri

3. Bölüm: Algoritmalar

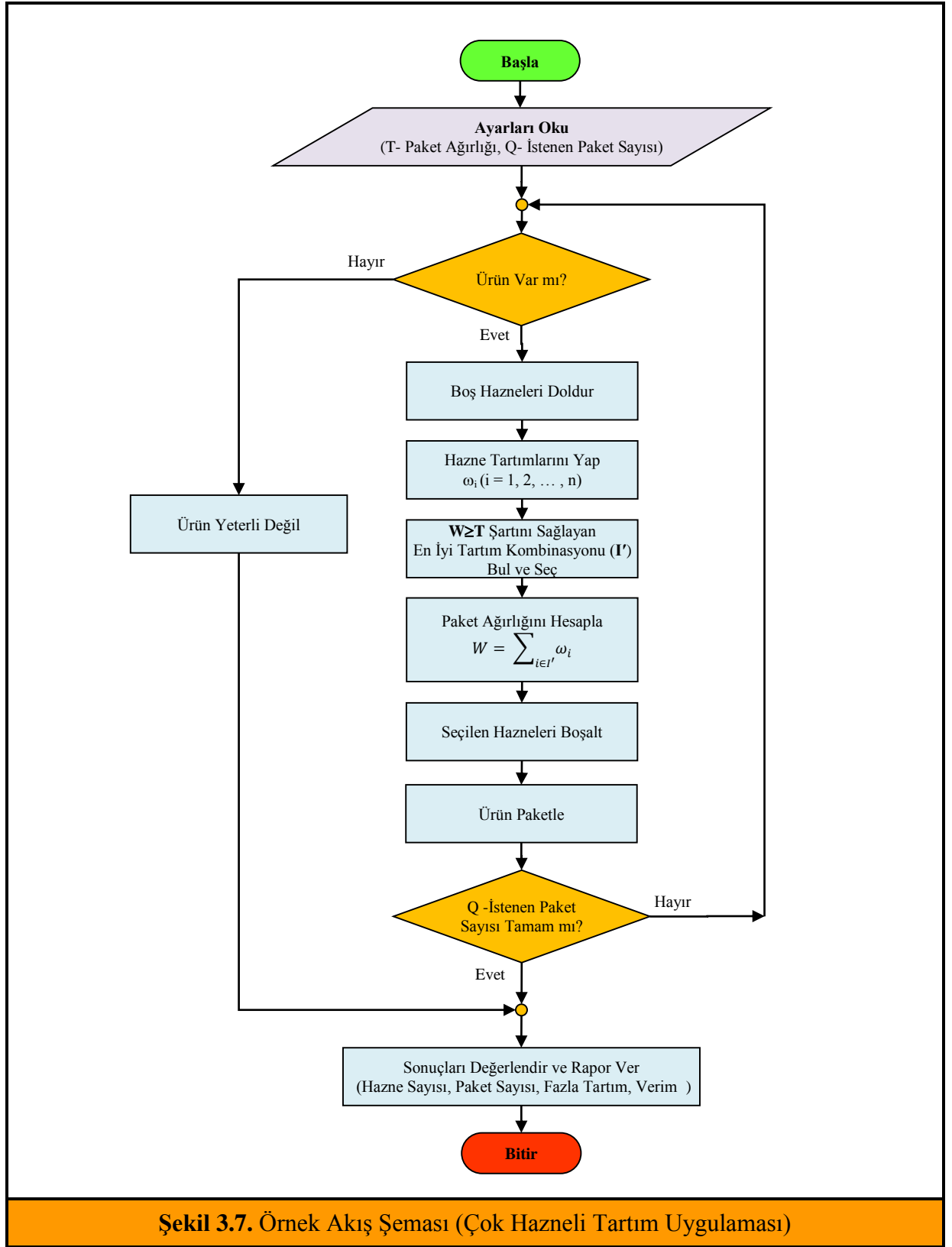
Şekil 3.5 - Şekil 3.7’de akış şemalarına örnekler görülmektedir.





Şekil 3.6. Örnek Akış Şeması (Robot Hareketi)

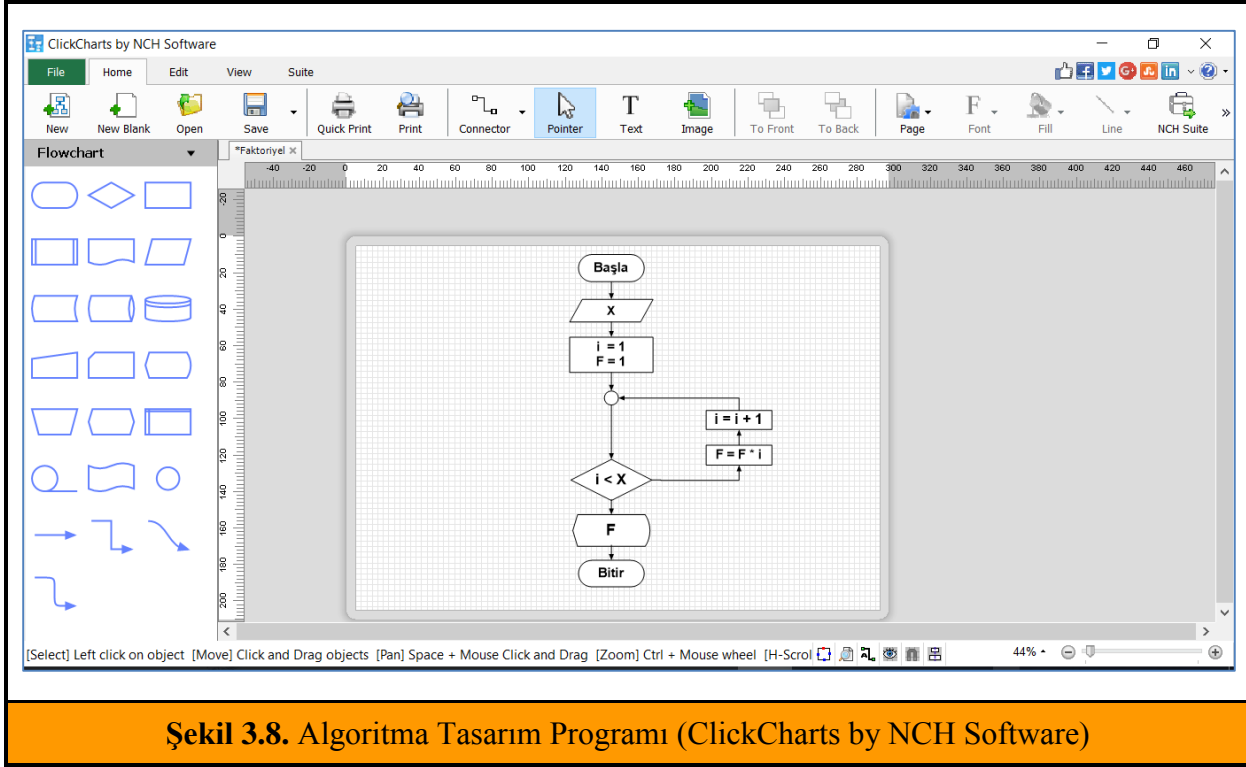
3. Bölüm: Algoritmalar



Şekil 3.7. Örnek Akış Şeması (Çok Hazneli Tartım Uygulaması)

Akış Şeması Hazırlama Araçları

Başta Microsoft Office paket programı olmak üzere çeşitli yazılım araçlarından yararlanarak akış şemalarının tasarlanması mümkündür. Şekil 3.8’de akış şeması hazırlamak amacıyla kullanılan **ClickCharts by NCH Software** yazılımı ekran görüntüsü görülmektedir. Bu yazılım <https://www.nchsoftware.com/chart/index.html> internet adresinden temin edilip, ticari olmayan çalışmalarda ücretsiz olarak kullanılabilir.



Şekil 3.8. Algoritma Tasarım Programı (ClickCharts by NCH Software)

3.6. Algoritma Uygulamaları

01. Algoritma Uygulaması

Toplama İşlemi

Klavyeden girilen iki tamsayının toplanarak ekrana yazdırılması

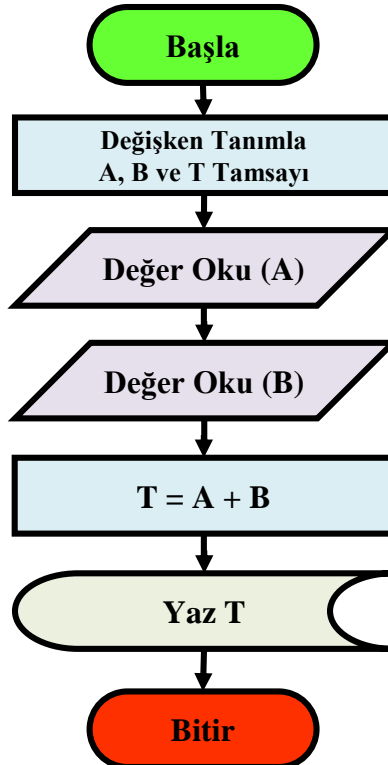
Doğal Dil

1. Adım : Başla
2. Adım : Birinci sayıyı gir
3. Adım : İkinci sayıyı gir
4. Adım : İki sayıyı topla
5. Adım : Toplam değeri yazdır
6. Adım : Bitir

Sözde Kod

1. Adım : BAŞLA
2. Adım : A, B ve T işaretli tamsayı değişkenlerini TANIMLA
3. Adım : A değerini OKU
4. Adım : B değerini OKU
5. Adım : $T = A + B$
6. Adım : T değerini YAZ
7. Adım : BİTİR

Akış Şeması

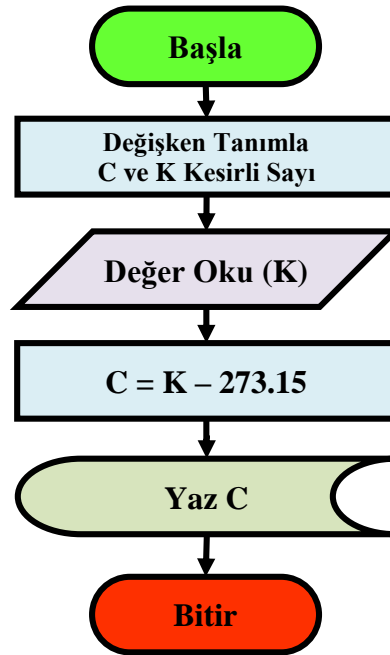


02.Algoritma Uygulaması**Sıcaklık Birimi Dönüşümü**

Klavyeden girilen **K-Kelvin** birimindeki sıcaklığın, **C-Celcius** birimine dönüştürülerek sonucun ekrana yazdırılması

Sözde Kod

- 01. Adım** : BAŞLA
02. Adım : C ve K değişkenlerini kesirli sayı olarak TANIMLA
03. Adım : K değerini OKU
04. Adım : $C = K - 273.15$ işlemini HESAPLA
05. Adım : C değerini YAZ
06. Adım : BİTİR

Akış Şeması

03.Algoritma Uygulaması

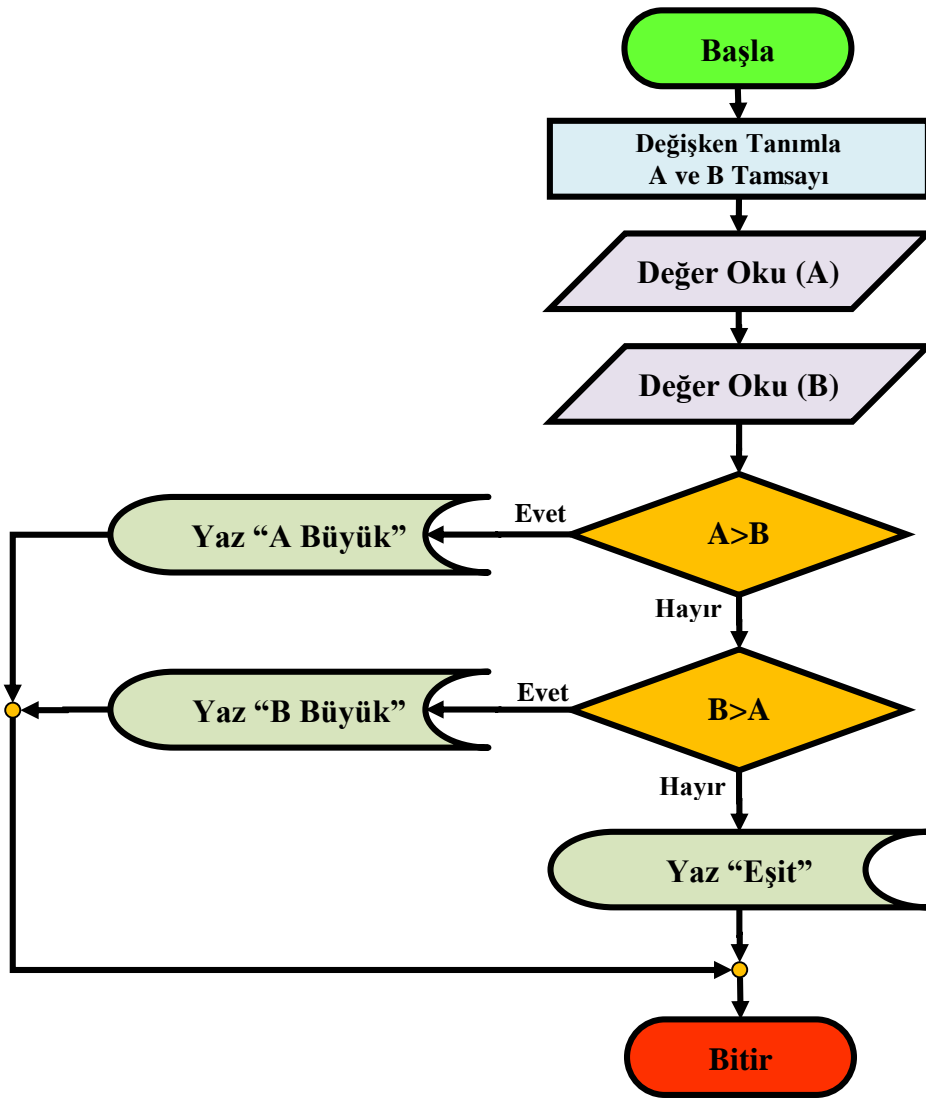
Karşılaştırma İşlemi

Klavyeden girilen iki tamsayıdan büyük olanın bulunarak ekrana yazdırılması

Sözde Kod

1. Adım : **BAŞLA**
2. Adım : A ve B değişkenlerini işaretli tamsayı olarak **TANIMLA**
3. Adım : A ve B değişkenlerini **OKU**
4. Adım : **EĞER** $A > B$ ise "A Büyüktür" **YAZ**
5. Adım : **DEĞİLSE EĞER** $B > A$ ise "B Büyüktür" **YAZ**
6. Adım : **DEĞİLSE** " $A=B$ " **YAZ**
7. Adım : **BİTİR**

Akış Şeması

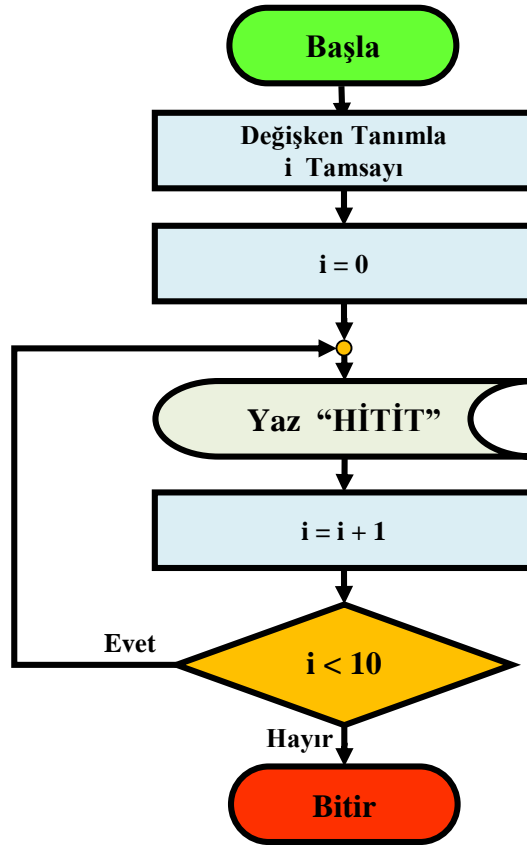


04. Algoritma Uygulaması**Çevrimsel Ekran Yazdırma İşlemi**

Ekran 10 kez "HİTİT" yazdırılması

Sözde kod

1. Adım : **BAŞLA**
2. Adım : i değişkenini işaretli tamsayı olarak **TANIMLA**
3. Adım : $i = 0$
4. Adım : Ekran "HİTİT" **YAZ**
5. Adım : $i = i + 1$
6. Adım : **EĞER** $i < 10$ ise 4. Adıma **GİR**
7. Adım : **BİTİR**

Akış Şeması

05. Algoritma Uygulaması

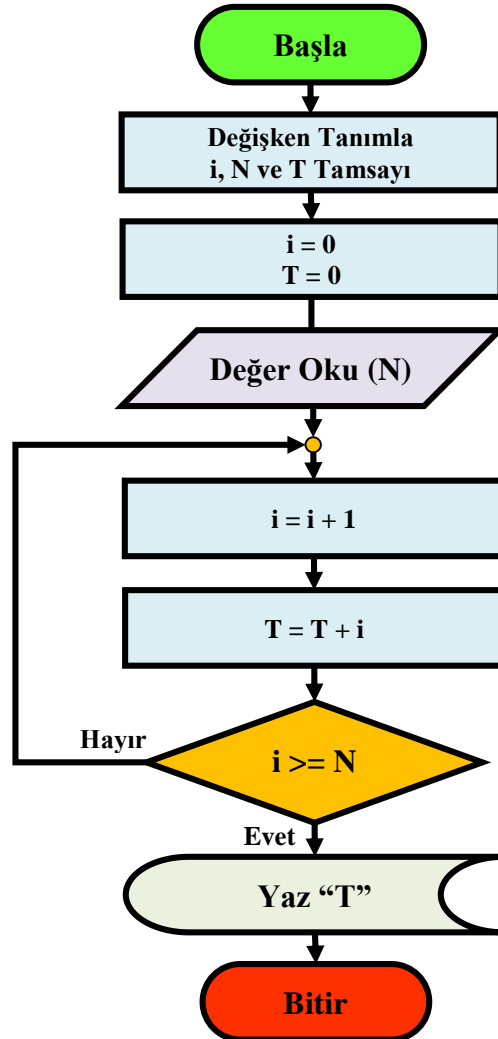
Seri Toplama İşlemi

1+2 ... + N seri toplama işlemi için, N değeri klavyeden girildiğinde sayıların toplamının bulunarak ekrana yazdırılması

Sözde Kod

1. Adım : **BAŞLA**
2. Adım : i, N ve T değişkenlerini işaretsiz tamsayı olarak **TANIMLA**
3. Adım : $i = 0$
4. Adım : $T = 0$
5. Adım : N değerini **OKU**
6. Adım : **EĞER** $i \geq N$ ise 10. Adıma **GİT**
7. Adım : $i = i + 1$
8. Adım : $T = T + i$
9. Adım : 6. Adıma **GİT**
10. Adım : T değerini **YAZ**
11. Adım : **BİTİR**

Akış Şeması



06. Algoritma Uygulaması**Seri Toplama İşlemi**Klavyeden girilen **N** adet sayının toplamını bulunarak ekrana yazdırılması**Sözde Kod**

1. Adım : **BAŞLA**
2. Adım : **i, N, X** ve **T** değişkenlerini işaretli tamsayı olarak **TANIMLA**
3. Adım : **i = 0**
4. Adım : **T = 0**
5. Adım : **N** değerini **OKU**
6. Adım : **EĞER** $i \geq N$ ise 11. Adıma **GİT**
7. Adım : **X** değerini **OKU**
8. Adım : **i = i + 1**
9. Adım : **T = T + X**
10. Adım : 6. Adıma **GİT**
11. Adım : **T** değerini **YAZ**
12. Adım : **BİTİR**

Akış Şeması