

CENG 111 ALGORİTMALAR VE PROGRAMLAMA

Doç. Dr. Tufan TURACI

tturaci@pau.edu.tr

- Pamukkale Üniversitesi
- Mühendislik Fakültesi
- Bilgisayar Mühendisliği Bölümü
- Hafta1-2

•Algoritma Kavramına Giriş

- Problem Çözme
- Algoritma Nedir?
- Algoritma Gösterim Şekilleri
 - Düz yazı
 - Sözde kod
 - Akış diyagramları
- Mantıksal Yapılar
- İşlemler ve Operatörler
- Algoritmada Kullanılan Yapılar

Problem Çözme

- İnsanlar günlük hayatta sürekli düşünme halindedir ve problemleri çözerler.
- Birçok problem, az ya da hiç düşünülmeden çözülebilir.

Problem: Bugün okula nasıl giderim?

Çözüm: Adım adım neler yapmalıyız?

Problemi Kim Çözer?

Bilgisayar, sadece kendisine söylenen çözüm yolunu nasıl yapacağını bilir.

Bilgisayara çözüm yolunu öğretmeliyiz....

Bilgisayara nasıl iş yaptırılır?

Bilgisayara nasıl iş yaptırılır, nasıl iletişim kurulur?

Bir “Program” ile. Bilgisayarlar program olmadan çalışmazlar.

Bilgisayarın dili **Makine Dilidir**. Onunla makine mantığı ile iletişim kurabiliriz.

Bu da **Algoritmalar** ile olur.

Problem Çözme Sırası

1. Problemi anlama (Understanding, Analyzing)
2. Bir çözüm yolu geliştirme (Designing)
3. Algoritma ve program yazma (Writing)
4. Tekrar tekrar test etme (Reviewing)

Algoritma Nedir?

- Tanım 1:** Belirli bir görevi yerine getiren sonlu sayıdaki işlemler dizisidir.
- Tanım 2:** Verilen herhangi bir sorunun çözümüne ulaşmak için uygulanması gerekli adımların hiç bir yoruma yer vermeksizin açık, düzenli ve sıralı bir şekilde söz ve yazı ile ifadesidir. Algoritmayı oluşturan adımlar özellikle basit ve açık olarak sıralanmalıdır.

- Algoritmanın etkin bir şekilde oluşturulması program yazma adımından çok daha önemlidir.
- Hazırlanan algoritmanın programlama diliyle yazılması işin basit kısmıdır.
- Tasarladığınız algoritma iyi değilse, kullandığınız dilin hiçbir önemi yoktur (C, C++, C#, Java, Visual Basic vb.)
- Bir sorunun çözümü için birbirinden farklı birden fazla sayıda algoritma hazırlanabilir. Bu da gösteriyor ki herhangi bir problemin çözümü için birbirinden farklı yüzlerce bilgisayar programı yazılabilir.

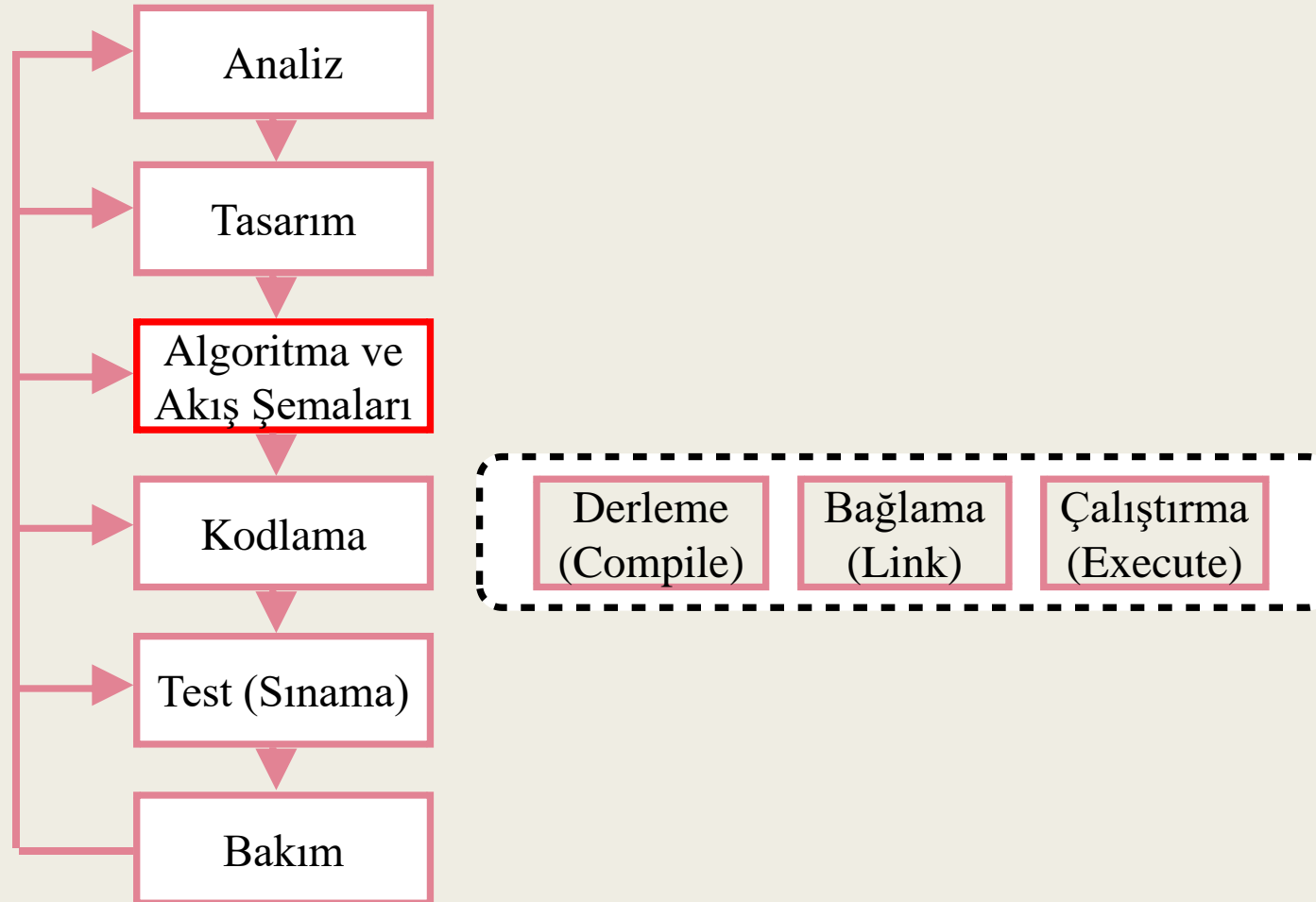
Algoritmaların Genel Özellikleri

- Giriş/çıkış bilgisi
- Sonluluk
- Kesinlik
- Etkinlik
- Başarım ve performans

Türlerine göre Algoritmalar

- Arama algoritmaları
- Sıralama algoritmaları
- Evrimsel algoritmalar
- Genetik algoritmalar
- Optimizasyon algoritmaları
- Veri madenciliği algoritmaları
- .
- .
- .

Yazılım Geliştirme Döngüsünde Algoritma Nerededir?



Algoritma Gösterim Şekilleri

1. Düz yazı ile gösterim
2. Sözde kod (pseudocode) ile gösterim
3. Akış diyagramı(şeması) ile gösterim

Örnek1: İki sayının toplamını bulan algoritmayı yazınız.

Düz Yazı İle:

1. BAŞLA
2. Birinci sayıyı gir
3. İkinci sayıyı gir
4. İki sayıyı topla
5. Sayıların toplam değerini yazdır
6. BİTİR

Sözde Kod İle:

Toplam için TOP, birinci sayı için A, ikinci sayı için B değerlerini ata

1. BAŞLA
2. A değerini OKU
3. B değerini OKU
4. $TOP = A + B$
5. TOP değerini yazdır
6. BİTİR

Örnek2: Bir dikdörtgenin alanını bulan algoritmayı yazınız.

Düz Yazı İle:

1. BAŞLA
2. Uzun kenarı gir
3. Kısa kenarı gir
4. Uzun kenar ile Kısa kenarı çarp
5. Çarpım değerini yazdır
6. BİTİR

Alan için A, uzun kenar için X, kısa kenar için Y değerlerini ata

1. BAŞLA
2. X değerini OKU
3. Y değerini OKU
4. $A = X * Y$
5. A değerini yazdır
6. BİTİR

Akış Diyagramları (Şemaları)

Bir algoritmanın görsel şekiller ve sembollerle ifade edilmiş haline **Akış Diyagramı** adı verilir.

Akış diyagramları herkes tarafından kolayca anlaşılır.



BAŞLA




BİTİR


Akış şemasının başlangıç ve bitiş yerlerini gösterir. Başlangıç simgesinden çıkış oku vardır. Bitiş simgesinde giriş oku vardır.



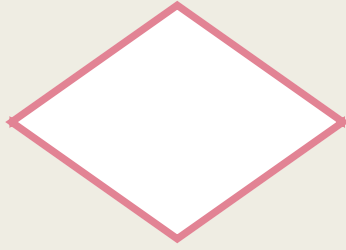
Aritmetik işlemler ve değişik atama işlemlerinin temsil edilmesi için kullanılır.



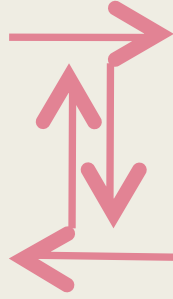
Dışarıdan bilgi giriş çıkışı için kullanılır.



Belgeye, yazıcıya, ekrana çıktı için kullanılır.



Kontrol ve karar verme işlemlerini temsil eder.



Oklar şemanın akış yönünü belirler.

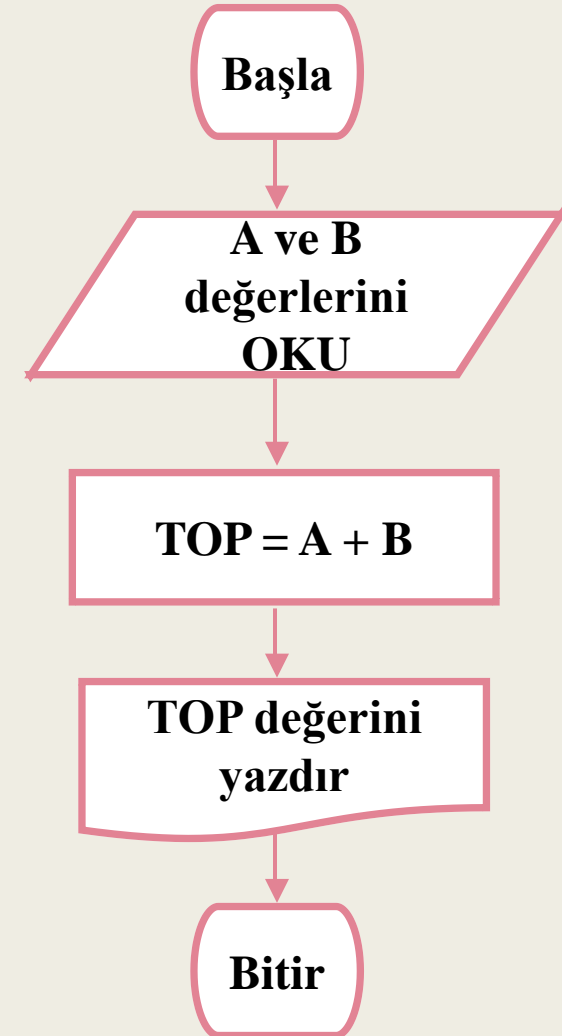
İki sayının toplamını bulan algoritmanın akış diyagramı

Akış Diyagramı:

Sözde Kod:

Toplam için TOP, birinci sayı için A, ikinci sayı için B seç

1. BAŞLA
2. A değerini OKU
3. B değerini OKU
4. $TOP = A + B$
5. TOP değerini yazdır
6. BİTİR



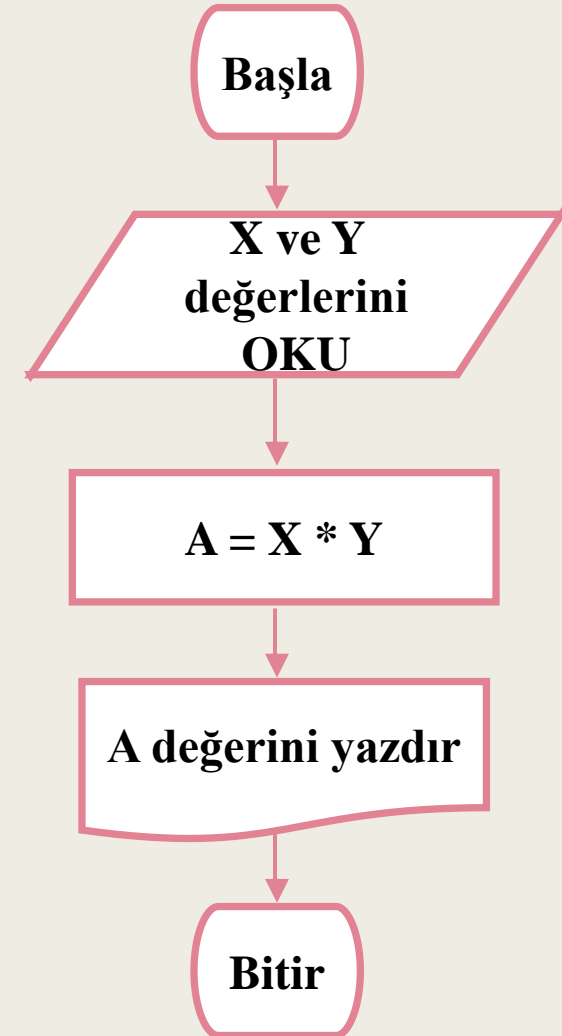
Bir dikdörtgenin alanı bulan algoritmanın akış diyagramı

Akış Diyagramı:

Sözde Kod:

Alan için A, uzun kenar için X, kısa kenar için Y değerlerini ata

1. BAŞLA
2. X değerini OKU
3. Y değerini OKU
4. $A = X * Y$
5. A değerini yazdır
6. BİTİR



Mantıksal Yapılar

- Bir bilgisayar programının geliştirilmesinde kullanılan programlama diline olursa olsun bu programların akış şemalarında genel olarak üç basit mantıksal yapı kullanılır.

1. Sıralı Yapı

2. Karar Verme Yapısı

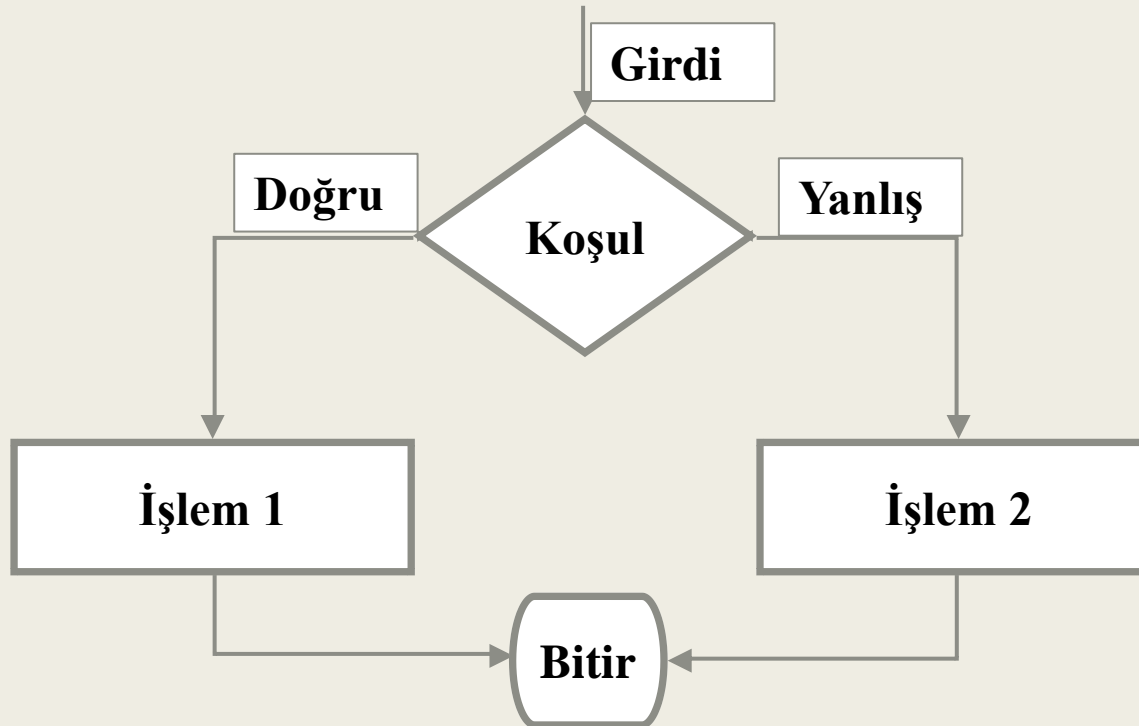
3. Tekrarlı Yapı

- Sıralı yapı, hazırlanacak programdaki her işlemin mantık sırasına göre nerede yer alması gerektiğini vurgular. Bu yapı sona erinceye kadar ikinci bir işlem başlayamaz.



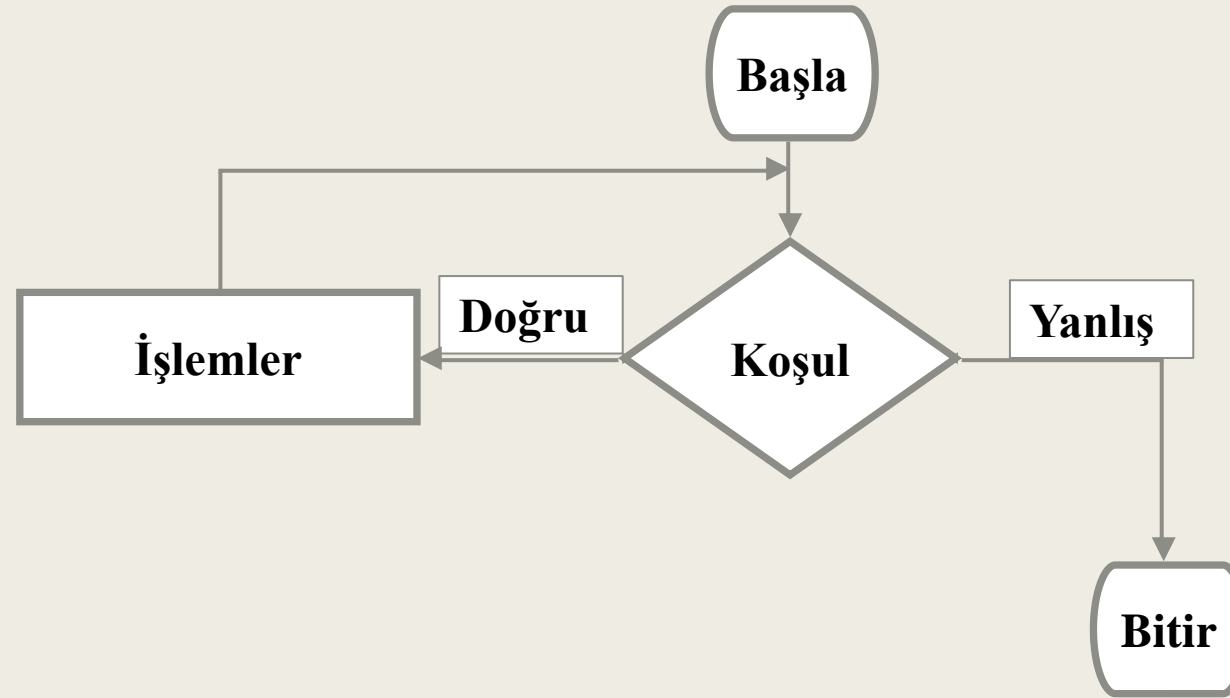
Mantıksal Yapılar: Karar Verme Yapısı

- Birden fazla sıralı yapı seçeneğini kapsayan modüllerde, hangi şartlarda hangi sıralı yapının seçileceğini belirler.



Mantıksal Yapılar: Tekrarlı Yapı

- Algoritma içinde, bazı satırlar tekrarlı şekilde işlem görüyorsa, bir döngü söz konusudur. Döngülere belirli bir koşul geçerli olduğu sürece devam eden eylemleri tanımlamak için başvurulur.



İşlemler ve Operatörler

- İşlemler 3'e ayrılır:

1. Matematiksel İşlemler

- Temel Aritmetik İşlemler: Toplama, çıkarma, çarpma, bölme.
- Matematiksel Fonksiyonlar: Üstel, logaritmik, trigonometrik, ...

2. Karşılaştırma İşlemleri

3. Mantıksal İşlemler

Matematiksel İşlemler

İşlem	Gösterim
Toplama	$a + b$
Çıkarma	$a - b$
Çarpma	$a * b$
Bölme	a / b
Üs alma	$a ^ b$

Matematiksel Yazım	Bilgisayar Gösterim
$2a + 3b - c^2$	$(2*a)+(3*b)-(c^2)$
$\frac{2ac^2 - b^3}{a + b}$	$((2*a*c^2)-(b^3))/(a+b)$
$c(a^2 + b^3) - 4\sqrt{b}$	$c*(a^2+b^3)-4*(b^(1/2))$

Karşılaştırma İşlemleri

- Değişkenlerin büyük olma, küçük olma ve eşit olma durumlarını kontrol eden işlemlerdir.

İşlem Sembolü	Anlamı
=	Eşittir
<>	Eşit değildir
>	Büyüktür
<	Küçüktür
>=	Büyük eşittir
<=	Küçük eşittir

Mantıksal İşlemler

- «Ve, Veya, Değil» operatörleri hem matematiksel işlemlerde hem de karar ifadelerinde kullanılır.

Mantıksal İşlem	Komut
Ve	And
Veya	Or
Değil	Not

- ve bağlacı ile söylenmek istenen her iki koşulun da sağlanmasıdır. VE bağlacı ile bağlanmış önermelerden en az birinin yanlış olması sonucu yanlış yapar.
- veya bağlacı ile bağlanan koşullardan bir tanesinin doğru olması sonucu doğru yapar.

Mantıksal İşlemler

- **değil** bağlacı; doğruyu yanlış, yanlış doğru yapar.

Örnek: Bir sınıftaki erkek öğrencilerden yaşı 25'in üzerinde olanları ekrana yazdır.

Eğer;

- (Cinsiyet = «Erkek») ve (Yas > 25) ise ekrana yazdır.

Algoritmada Kullanılan Yapılar

1. Değişken
2. Atama
3. Sayaç
4. Döngü

Algoritmada Kullanılan Yapılar:

Değişken

- Programın her çalıştırılmasında, farklı değerler alan bilgi/bellek alanlarıdır.
- Programcı tarafından oluşturulur.
- Programdaki değişkenleri, sabitleri, kayıt alanlarını, özel bilgi tiplerini vb. adlandırmak için kullanılan kelimelerdir.
- İngiliz alfabesindeki A-Z veya a-z arasındaki 26 harf ile 0-9 arası rakamlar kullanılabilir.
- Değişkenler sadece alt çizgi (_) kullanılabilir.
- Değişkenler harfle veya alt çizgiyle başlayabilir.
- Değişkenler, rakamla başlayamaz veya sadece rakamlardan oluşamaz.

•Örnekler:

Değişken olabilir:

s

a5

adsoyad

ad_soyad

Değişken olamaz:

123

2a

b,2

Algoritmada Kullanılan Yapılar :

Atama

Sağdaki **Değer** sonucu **Değişken**'e aktarılır.

Bu durumda değişken'in bir önceki değeri varsa silinir.

s = 0

enb=s1

Algoritmada Kullanılan Yapılar :

Sayaç

- Belirli sayıda işlemin yaptırılması ve üretilen değerlerin sayılması durumunda kullanılır.
- Bu tür sayma işlemlerine algoritmada Sayaç adı veriler.
- Sayaçlar da birer değişkendir.

Örnek:

```
s = s + 1  
top=top+s
```

Algoritmada Kullanılan Yapılar :

Döngü

- Algoritmalarda bazı işlemler, belirli ardışık değerlerle gerçekleştirilmekte veya belirli sayıda yaptırılmaktadır.
- Programlardaki belirli işlem bloklarını, verilen sayıda gerçekleştiren işlem akış çevrimlerine “**döngü**” denir.
- 1 ile 99 arasındaki tek sayıların toplamını hesaplayan programda

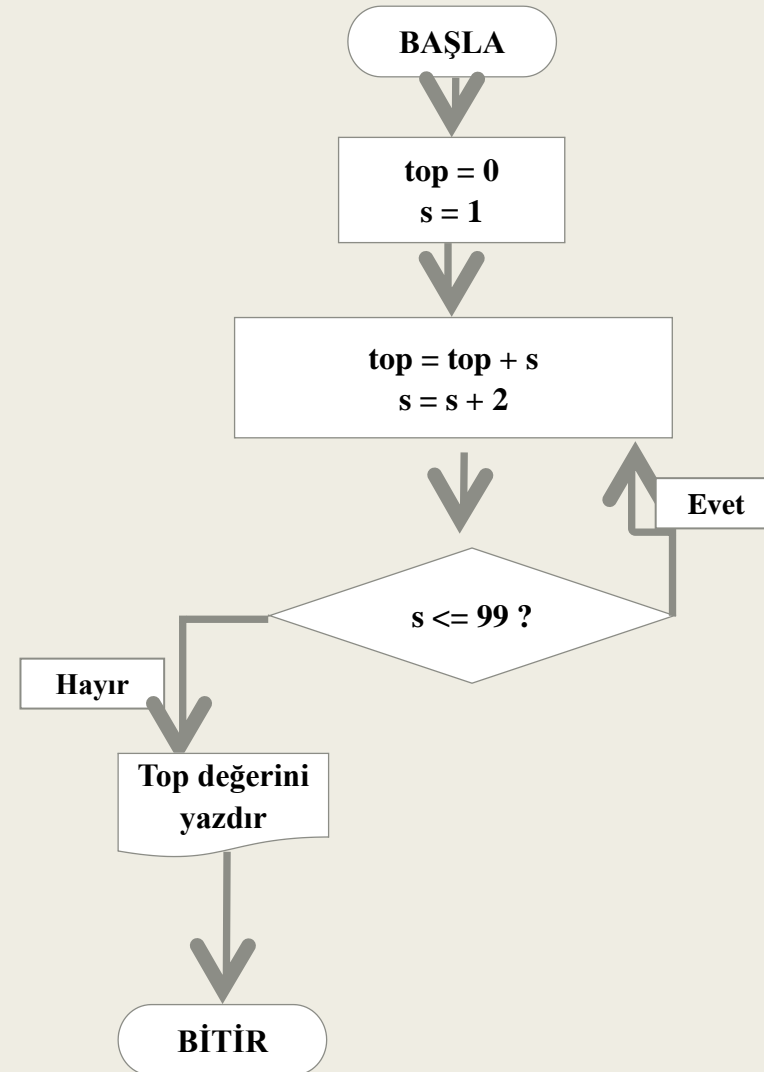
$$\text{top}=1+3+5 \dots$$

yerine 1 ile 99 arasında ikişer artan bir döngü kurulur ve döngü değişkeni ardışık toplanır.

Sözde Kod

1. BAŞLA
2. $top = 0$
3. $s = 1$
4. $top = top + s$
5. $s = s + 2$
6. Eğer $s \leq 99$ ise Adım 4' e git.
7. top değerini yazdır.
8. BİTİR

Akış Diyagramı



Örnek: 1-10 arasındaki sayıların ekrana yazdırma

1. BAŞLA
2. $s = 1$
3. s değerini yazdır
4. $s = s + 1$
5. Eğer $s < 11$ ise Adım 3'e git.
6. BİTİR

Değişken İzleme Tablosu

Eski s değeri	Yeni s değeri	Ekran
1	2	1
2	3	2
3	4	3
...
10	11	10

Soru1: Klavyeden girilen 50 adet tamsayı içinde çift sayı olanların ortalamasını bulan algoritmayı tasarlayınız ve akış diyagramını çiziniz.

Soru2: Klavyeden girilen 5 adet tamsayının faktöriyelerinin toplamını bulan algoritmayı tasarlayınız ve akış diyagramını çiziniz.

Soru3: $\sum_{s=1}^{10} \sum_{k=1}^{15} (3s^2 + 5k - 7)$ toplamını bulan algoritmayı tasarlayınız ve akış diyagramını çiziniz.

Kaynaklar

- C: How to Program Third Edition Harvey M. Deitel ; Paul J. Deitel.
- C Programlama Dili Dr. Rıfat Çölkesen Papatya Yayıncılık.
- Problem Solving and Program Design in C, 7/E Jeri R. Hanly; Elliot B. Koffman.
- C Programlama dili; İbrahim Güney; Nobel Yayıncılık.
- Algoritma Geliştirme ve Programlamaya Giriş, Fahri Vatansever, Seçkin yayıncılık
- C Programlama Ders Notları, A. Kadir YALDIR, Pamukkale Üniversitesi ders notları.