



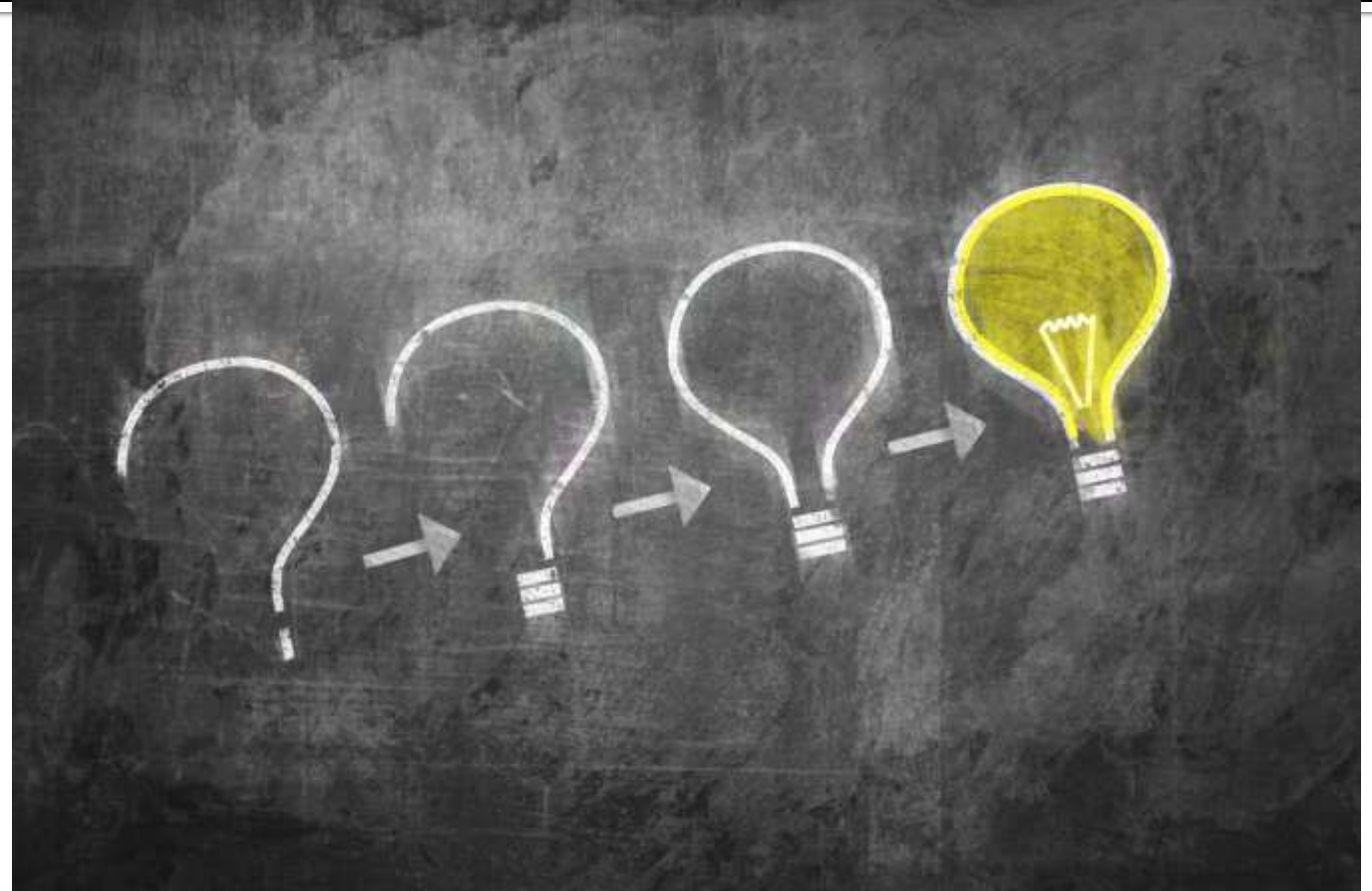
PROBLEM ÇÖZME SÜRECİ

PROBLEM NEDİR?



- Problem; günlük hayatımızda karşılaştığımız, çözüm aranması gereken ve çözümü için bilgi, mantık, deneyim ya da dikkat isteyen durumları ifade eder.
- Örneğin; Gece geç yattığınızdan dolayı sabah okul servisini kaçırmamız ve okula geç gitmemiz bir problemdir.

- Gnlk hayatınızda siz ne gibi problemlerle karřılařıyorsunuz?



- Yemek yapacaksınız ama evde tuz kalmamış.
- Sabah kalktın ve giyinmen gerekiyor, kıyafetini nasıl seçersin?
- Sabah geç uyandınız ve servisi kaçırdınız.
- Otobüse binmek üzeresiniz ve kartınızı unuttuğunuzu fark ettiniz.
- Yolda yürürken ayakkabınızın topuğu kırıldı.

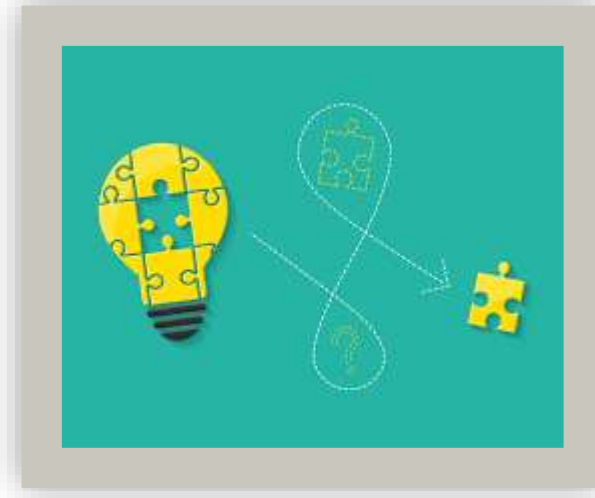
PROBLEM ÇÖZME TEKNİKLERİ

- Programlama sürecinde de problemin çözümüne yönelik yol ve yaklaşımları belirlemek gerekir ama öncelikle genel kural ve teknikleri bilmek yararlıdır.
- Bazı genel kurallar neredeyse tüm problemler için kullanılabilir



GENEL KURALLAR

- ✓ Her Zaman Bir Planınız Olsun
- ✓ Problemi Tekrar İfade Edin
- ✓ Problemi Küçük Parçalara Ayırın
- ✓ Önce Bildiklerinizden Yola Çıkın
- ✓ Problemi Basitleştirin
- ✓ Benzerlikleri Arayın
- ✓ Deneme Yapın
- ✓ Asla Vazgeçmeyin



i. Her Zaman Bir Planınız Olsun

- Belirsiz bir durumu yaşamak yerine her zaman bir planınız olmalıdır. Bu, en önemli kuraldır.
- Belki oluşturduğunuz çözüm planı ilk denemelerde sonuç vermeyecek ama her seferinde sizi çözüme bir adım daha yaklaştıracak ipuçları elde etmenizi sağlayacaktır. Denediğiniz yaklaşım ne olursa olsun, fikir üretmemekten ve deneme yapmamaktan her zaman daha iyidir. Planlama yapmak aynı zamanda hedef belirlemek ve bu hedefe ulaşmak anlamına gelir.

ii. Problemi Tekrar İfade Edin

- Bazen problemi tekrar ifade etmek, göremediğimiz bir ayrıntıyı görmemizi ya da problemi daha kolay çözmek adına bir ipucu yakalamamızı sağlayabilir. Problemi farklı biçimlerde sunmak çözüm sürecine ışık tutmasa bile bazen yalnızca problemi doğru anlayıp anlamadığımızı teyit etme açısından önemlidir.

iii. Problemi Küçük Parçalara Ayırın

- Verilen problemi adımlara ya da bölümlere ayırmak, çözümü kolaylaştıracaktır. Bir problemi iki bölüme ayırdığımız düşünüldüğünde, her bir parçanın çözümünün tümünü çözmeye göre yarı yarıya kolaylaştığını düşünebiliriz.
- Bir problemi çözülebilir küçük parçalara bölmek, çözüm işlemini kolaylaştırır ve hızlandırır.

iv. Önce Bildiklerinizden Yola Çıkın

- Programlama yaparken öncelikle bildiklerimiz ile başlamalı ve sonra yeni çözümler arayışına girmeliyiz. Problemi küçük parçalara bölerek çözebildiğiniz parçadan başlamalısınız. Bu parçaları çözerken diğer parçalarla ilgili olarak aklınıza yeni fikirler geldiğini ve aynı zamanda kendinize olan güvenin arttığını göreceksiniz.

v. Problemi Basitleştirin

- Çözmekte zorlandığınız bir problemle karşılaşırsanız problemin kapsamını daraltmayı deneyin. Temel amacınız problemi basitçe ifade etmeye çalışmak olmalıdır. Çözüm için denediğiniz yaklaşımlar, size gerçek çözüm için yol gösterecektir. Problemi basitleştirmek size aslında problemdeki zorluğun neden kaynaklandığını da gösterecektir.

vii. Deneme Yapın

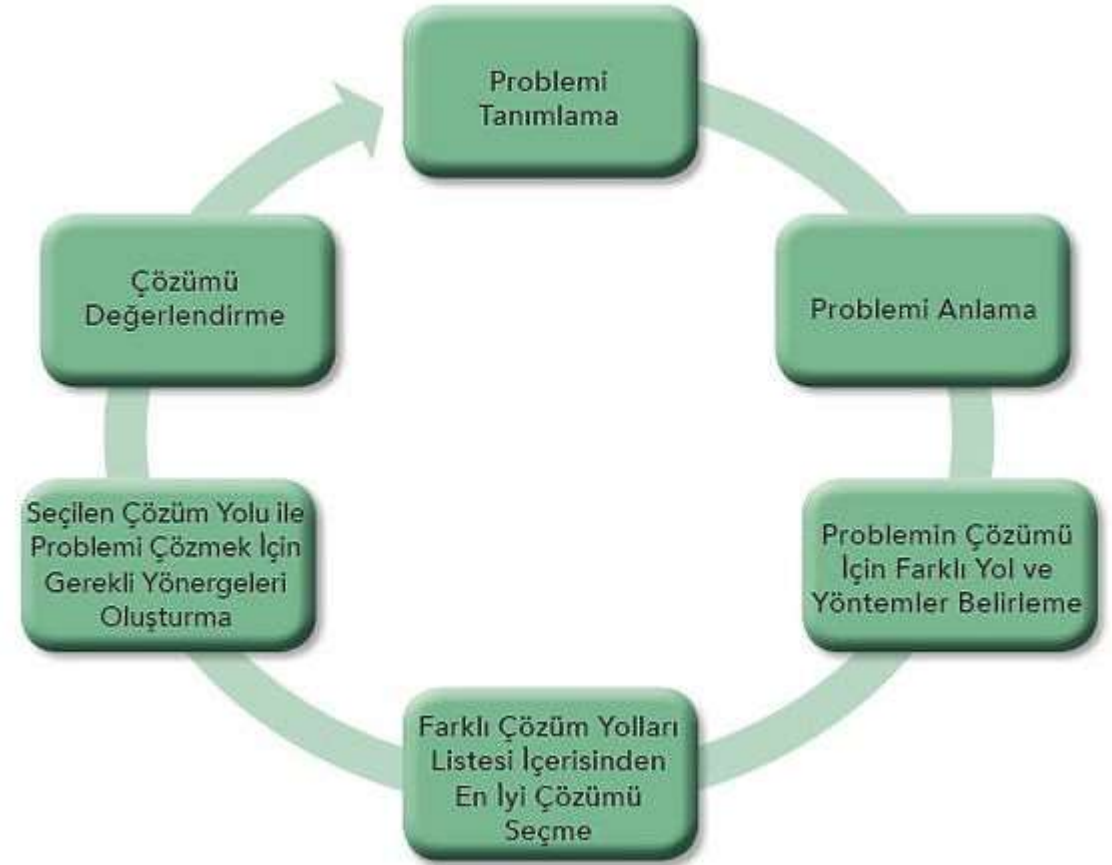
- Bazen bir problemi çözenin en kolay yolu denemek ve sonuçlarını gözlemlemektir. Bu tahmin etmekten çok farklıdır. Bir çözümü tahminen öngörmek ile deneyip gözlemlemek farklı sonuçlar verir. Böylece problemi çözebilmek için gereken ipuçlarını elde edebilirsiniz.

viii. Asla Vazgeçmeyin

- Kararlılık, güven ve istek kaybolduđu zaman açık düşünemezsiniz, işlemler olması gerektiğinden uzun sürer ve gittikçe zorlaşır. Hatta öfke ve kızgınlığa bile dönüşebilir. Bu noktada moralimizin bozulmasına izin vererek başarısız olmak için bahane üretmiş oluruz. Böyle durumlar en etkili çözüm ara vermektir. Problemden tamamen uzaklaşarak geçirilecek vakit sonrasında daha verimli çalıştığınızı göreceksiniz.

PROBLEM ÇÖZME ADIMLARI

- Problemi Tanımlama
- Problemi Anlama
- Problemin Çözümü için Farklı Yol ve Yöntemler Belirleme
- Farklı Çözüm Yolları Listesinden En İyi Çözümü Seçme
- Seçilen Çözüm Yolu ile Problemi Çözmek için Gerekli Yönergeleri Oluşturma
- Çözümü Değerlendirme



1.PROBLEMI TANIMLAMA:

- Problemi çözmeye başlamadan önce problemin açık, anlaşılır ve çok doğru bir şekilde tanımlanmış olması gerekir.
- Problemin ne olduğunu bilemezseniz onu çözemezsiniz.



2. PROBLEMİ ANLAMA

- Problemin kaynağının ne olduğu belirlenmelidir. Bir problem ne kadar iyi anlaşılırsa çözümü o kadar kolay olacaktır.
- Bu konuda klasik ve önemli bir söz vardır: “Problemi anlamak, problemi yarı yarıya çözmek demektir.”

3.PROBLEMİN ÇÖZÜMÜ İÇİN FARKLI YOL VE YÖNTEMLER BELİRLEME:

- Problemin çözümü için olabildiğince farklı yol ve yöntem belirlemeli ve bu listenin, tüm olasılıkları içerdiğinden emin olmalısınız.
- Bunun için konu hakkında farklı kişilerin görüşlerini alabilirsiniz. Problem çözmek için tek bir yol yoktur; pek çok yol vardır.

4. FARKLI ÇÖZÜM YOLLARI LİSTESİ İÇERİSİNDEN EN İYİ ÇÖZÜMÜ SEÇME

- Bu adımda her bir çözümün olumlu ve olumsuz yönlerini ortaya koymalısınız. Bir önceki adımda belirlenen alternatifler arasından en uygun olanının seçilmesi gerekir.
- Problem çözmek için tek bir yol yoktur; en iyi yol vardır

5. SEÇİLEN ÇÖZÜM YOLU İLE PROBLEMI ÇÖZMEK İÇİN GEREKLİ YÖNERGELERİ OLUŞTURMA:

- Bu adımda numaralandırılmış ve adım adım yönergeler oluşturmanız gerekir.



6. ÇÖZÜMÜ DEĞERLENDİRME:

- Çözümü test etmek ya da değerlendirmek, sonucun doğruluğunu kontrol etmek anlamına gelir.
- Sonucun doğru olması ve problemi olan bireyin beklentilerini karşılama düzeyi önemlidir.
- Sonuç yanlış çıkmış ya da bireyin beklentilerini karşılamamış ise problem çözme sürecine baştan başlamak gerekir.

PROBLEM TÜRLERİ

- Problemlerin her zaman sıradan çözümleri olmaz. Kek yapmak ya da araba kullanmak gibi problemleri çözmek için bir dizi eylem gerekir. Adım adım yönergelere dayalı olan bu çözümlere **“algoritmik çözümler”** denir.
- En iyi yolu seçtikten sonra sonuca, ilgili adımları izleyerek ulaşılır. Bu adımlardan oluşan yapıya **“algoritma”** denir.
- En lezzetli ekmeği seçmek ya da işleri büyütme için yatırım yapmak gibi problemlerin ise açık ve net ifade edilen yanıtları yoktur. Bu çözümler bilgi ve deneyim gerektirir, bir dizi deneme ve yanılma sürecinden oluşur. Doğrudan işlem adımları ile ulaşamayan sonuçlara **“keşfe dayalı çözümler”** denir.

BİLGİSAYARLAR İLE PROBLEM ÇÖZME

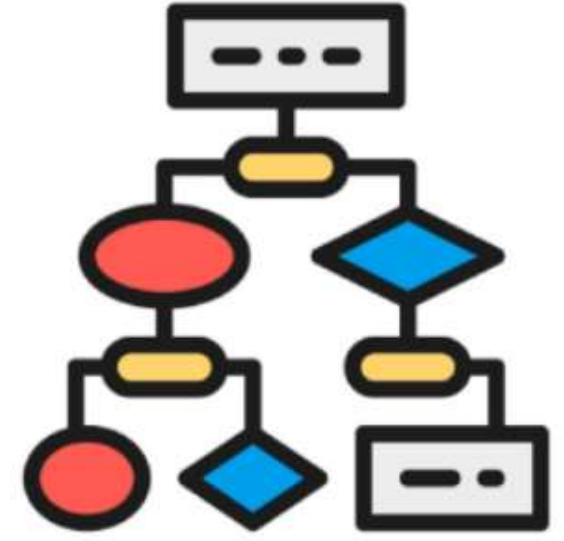
- Bu ders kapsamında “**çözüm**” demek problem çözme sürecinin 5. adımında yer alan işlem adımları ya da yönergeler anlamına gelmektedir.
- “**Sonuç**” demek, çıktı ya da tamamlanmış bilgisayar destekli yanıt demektir.
- “**Program**” ise herhangi bir bilgisayar dilinde kodlanmış, çözümü oluşturan işlem adımlarının tamamını ifade etmektedir.

- Bilgisayarlar, zor ve zaman alıcı olabilen **algoritmik çözümler** ile ilgilenmek üzere tasarlanmıştır.
- İnsanlar, **keşifsel çözümleri** bulma konusunda daha iyidirler ancak bilgisayarların çözebildiği ileri düzey hesaplama ve karmaşık problemleri çözme konusunda bilgisayarların hızlarına ulaşamazlar.
- Bilgisayarlar, üst düzey matematik problemlerini kolayca çözebilir ancak konuşmak ya da top atmak gibi davranışları yapamaz.
- **Bu tür işlemleri bilgisayarların anlayacağı bir dizi adım şekline nasıl dönüştürebiliriz?**

- Keşifsel problem türleri ile ilgilenen bilgisayar dalına “yapay zekâ” adı verilmektedir.
- Yapay zekâ uygulamaları, bilgisayarlara mevcut bilgileri kullanarak yeni bilgiler inşa etmesini sağlamaktadır. Böylece bilgisayarın problem çözme becerileri insanların yeteneklerine daha çok benzemektedir.
- Yapay zekâ özellikle robotik uygulamaları ile son yıllarda popüler bir çalışma alanı olmuştur. Bilgisayarlar insanlar gibi düşünmeye başlayana kadar daha çok algoritmik problemlerin çözüm süreçlerinde kullanılacaktır.

Algoritma Kavramı

- Algoritma kelimesi bir İslam Bilgini olan **El-Harezmi'nin** isminin Latince karşılığından gelmektedir. El-Harezmi matematik, gök bilim ve coğrafya alanlarında çalışmış, cebirin temelini oluşturmuş, bugünkü bilgisayar bilimi ve elektroniğin temeli olan 2'lik (binary) sayı sistemini ve 0'ı (sıfır) bulmuş önemli bir bilim insanıdır.
- Programlamanın öğrenilebilmesi için öncelikle algoritmanın ne olduğuna ve nasıl geliştirilmesi gerektiğine cevap bulunmalıdır.
- Problem çözme yöntemlerinden biri olan algoritma geliştirmek kodlamaya atılan ilk adımdır. Algoritma mantığı iyice kavrandıktan sonra bu mantık ile birlikte bir programlama dili kullanılarak yazılım geliştirme süreci başlar.



Algoritma Kavramı

- **Algoritma**, belirli bir mantığı olan, farklı düşünebilmeyi ve problem çözmeyi öğretmek için tasarlanan bir yoldur. Başka bir ifadeyle bir problemi çözmeye giden yolun basit, net ve belirli bir sıraya göre tasarlanmış hâlidir.

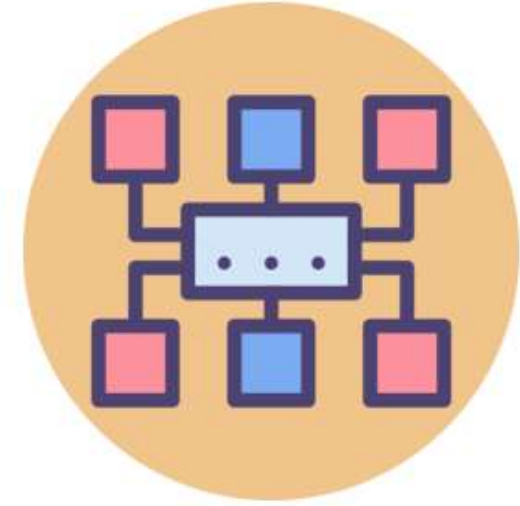
1. Açık ve net olmalıdır.

2. Kullanılacak olan girdiler iyi tanımlanmış olmalıdır

3. Çıktılar açık ve anlaşılır olmalıdır.

4. Algoritmalar hızlı olmalıdır.

5. Sonlu ve uygulanabilir olmalıdır.

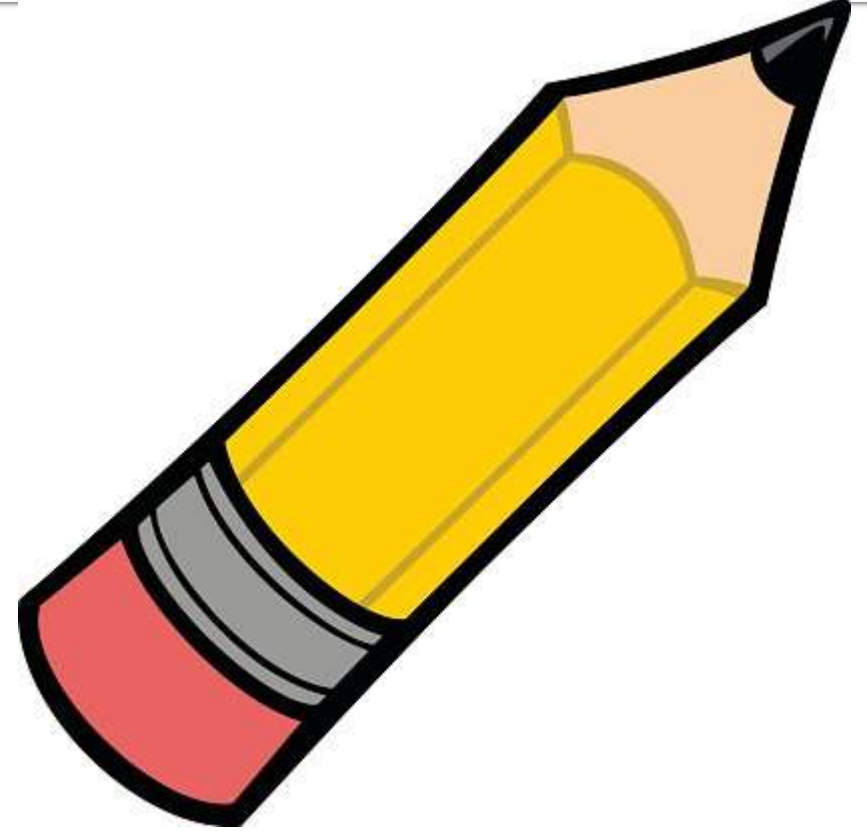


- Algoritmaların program haline getirilmesi için programlama dilleri kullanılır.
- Programlama dilleri kullanılarak yazılımlar geliştirilir.

Algoritma Örnekleri

- Günlük yaşamda karşılaştığımız problemleri bilerek veya farkında olmadan adım adım çözmeye çalışırız.
- Örneğin yazı yazarken kaleminizin ucu kırıldığında hangi adımları takip ederek bu sorunu çözersiniz.

- 1.Kalem tıraşı çıkar.
- 2.Kalemi al.
- 3.Çöp kovasının yanına git.
- 4.Kalemin ucunu aç.
- 5.Sırana geri dön.
- 6.Yazmaya devam et.



- Sabah kahvaltı için ay demleyeceksiniz. Adım adım ay demleme algoritmasını nasıl olur?



- Adım 1: Başla.
- Adım 2: Mutfağa git.
- Adım 3: Çaydanlığa su koy.
- Adım 4: Çaydanlığı ocağa koy.
- Adım 5: Ocağı yak.
- Adım 6: Su kaynadı mı, kontrol et.
- Adım 7: Cevap evet ise demliğe çay koy ve demle. Cevap hayır ise adım 6'ya git.
- Adım 8: 10 dakika bekle.
- Adım 9: Çayı doldur.
- Adım 10: Bitir

Yoğurt yapan algoritmayı yazalım

Adım 1: Başla

Adım 2: Yoğurdu kaba koy.

Adım 3: Su ekle.

Adım 4: Çırp.

Adım 5: Tuz koy.

Adım 6: Bardağa doldur.

Adım 7: Bitir.



Arabayı alıřtırıp yola ıkalım

- Adım 1: Bařla
- Adım 2: Sürücü koltuđuna ge.
- Adım 3: Emniyet kemerini tak.
- Adım 4: Aynaları kontrol et.
- Adım 5: Anahtarı tak.
- Adım 6: Kontakđı evir.
- Adım 7: El frenini indir.
- Adım 8: Vitese ge.
- Adım 9: Gaza bas.
- Adım 10: Bitir.



Algoritma Gösterim Şekilleri

- İki Sayının Toplamını Yapan Algoritma
- Düz Yazı İle Gösterim

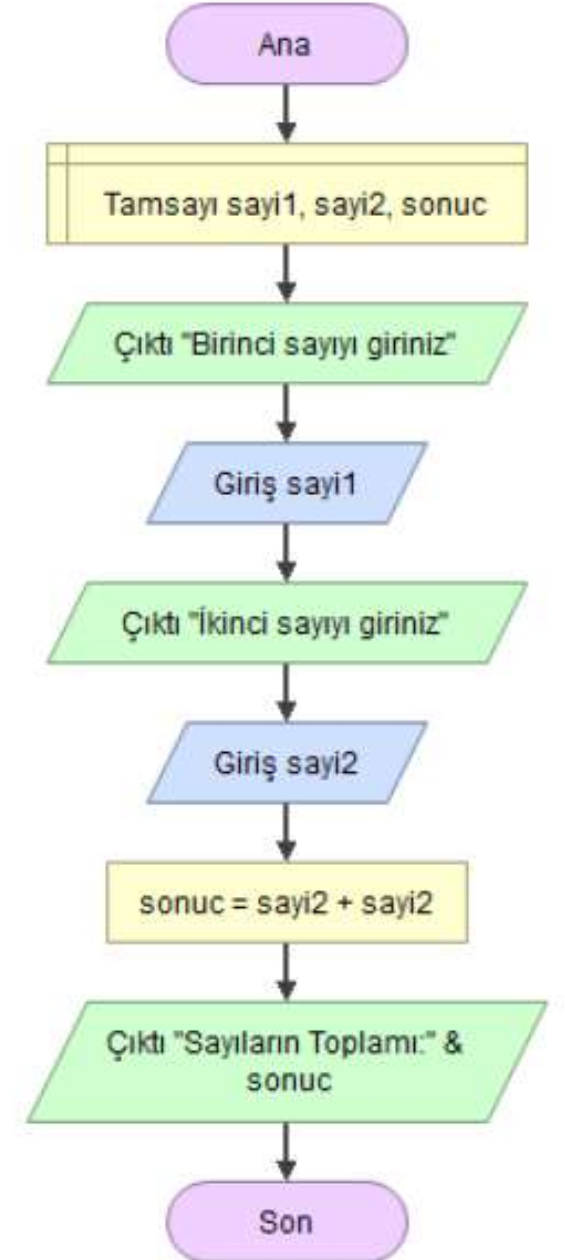
- Adım1: Başla
- Adım2: Birinci Sayıyı Oku
- Adım3: İkinci Sayıyı Oku
- Adım4: Birinci Sayı ile İkinci Sayıyı Topla
- Adım5: Toplamı Ekranı Yaz
- Adım6: Bitir

Algoritma Gösterim Şekilleri

- Pseudo Cod (Sözde Kod) ile Gösterim
- A₁: Başla
- A₂: sayi₁ oku
- A₃: sayi₂ oku
- A₄: sonuc=sayi₁+sayi₂
- A₅: Yaz sonuc
- A₆: Bitir

Algoritma Gösterim Şekilleri

- Akış Diyagramı ile Gösterim



Algoritma Gösterim Şekilleri

1. Düz yazı ile gösterim,
2. Sözde kod(pseudo code) ile gösterim,
3. Akış şeması ile gösterim.

Düz Yazı İle Gösterim

Adım1 : Başla

Adım2 : Birinci Sayıyı Oku

Adım3 : İkinci Sayıyı Oku

Adım4 : Birinci Sayı ve İkinci Sayıyı Topla

Adım5 : Toplamı Ekrana Yaz

Adım6 : Bitir

Pseudo Code(Sözde Kod) ile Gösterim

A1 : Başla

A2 : sayi1 oku

A3 : sayi2 oku

A4 : $sonuc = sayi1 + sayi2$

A5 : Yaz sonuc

A6 : Bitir

Akış Şeması İle Gösterim



Algoritma Örnekleri

- İki sayının ortalamasını bulan algoritma
(Düz yazı – Pseudo Cod)
- Girilen iki sayıdan birinci girilen sayıyı ikinci girilen sayıya bölüp sonucu ekrana yazdıran algoritma
(Düz yazı – Pseudo Cod)
- Girilen üç sayının çarpımını bulan algortima
(Düz yazı – Pseudo Cod)