



ÖĞRENME BİRİMİ 1

PROBLEM ÇÖZME VE ALGORİTMALAR



Neler Öğreneceksiniz?

Bu öğrenme birimi ile;

- Problem çözmenin önemini kavrayacak,
- Problem çözme kavramlarını bilecek,
- Probleme çözüm üretirken problem çözme aşamasındaki adımları izleyebilecek,
- Problem çözmede kullanılacak aritmetiksel ve mantıksal operatörler ile karar operatörlerini listeleyebilecek,
- İşlem önceliğini kavrayabilecek,
- Problem çözmek için algoritmaları kullanabilecek,
- Doğrusal, döngüsel ve mantıksal algoritmalar hazırlayabilecek,
- Problem çözmede algoritma hatalarını bulup düzeltebilecek,
- Problemin çözümü için gerekli algoritma ve akış şemalarını doğrusal, döngüsel yapıları ve karar mantık yapılarını kullanarak hazırlayabileceksiniz.

Anahtar Kelimeler:

Problem, problem çözme, algoritma, operatör, değişken, mantıksal algoritma, döngüsel algoritma, algoritma hatası, akış şeması.

Hazırlık Çalışmaları

1. Günlük hayatta ve bilişimde karşınıza çıkan çeşitli problemlerin neler olabileceğini araştırıp sınıfta arkadaşlarınızla paylaşınız.
2. Kodlama öğrenmenin önemini araştırınız.

1. PROBLEM ÇÖZME VE ALGORİTMALAR

1.1. Problem Çözme ve Temel Kavramlar

İnsanlık, varoluşundan beri çeşitli problemlerle karşılaşmış ve bu problemlere çözüm üretmeye çalışmıştır. Örneğin bilgiyi kalıcı hâle getirmek için yazıyı bulmuştur. Mağara duvarlarına ya da kil tabletlere yazılan çivi yazıları bilginin kalıcılığını sağlamış ancak taşıma güclüğü ortaya çıkmıştır. Bu problemi çözmek için papirüsler¹ icat edilmiştir. Daha sonra karşılaşılan her probleme yeni çözümler geliştirilerek günümüzdeki kâğıt ve kalemler üretilmiştir.

Çözülmesi gereken problemlerin geçmişte kaldığını söylemek mümkün değildir. İnsanoğlu yaşam boyu çeşitli problemlerle karşılaşır. Bu problemleri çözmeye becerisi kazanmak oldukça önemlidir. Bu bölümde öncelikle problemin ne olduğundan ve problemi çözerken nasıl bir yol izlenmesi gerektiğinden bahsedilecektir.

Günümüzde birçok problemi çözmek için bilgisayarlardan faydalanılır. Ancak bilgisayarlar problemi nasıl çözeceğini bilmez. Bir bilgisayarın problem çözebilmesi için işlem adımlarının belirlenmesi ve bunun bilgisayara yüklenmesi gerekmektedir. Algoritmalar bu sürecin ilk adımını oluşturur. Bölümün ilerleyen kısımlarında algoritmalar ve algoritmaların şekillerle ifade edildiği akış diyagramları anlatılacaktır.

1.1.1. Problem

Karşılaşılabilecek soruna veya çözülmesi gereken duruma problem denir. Örneğin; sınav esnasında kalem ucunun bitmesi, internetten alınan ayakkabının küçük gelmesi, okula geç kalınması, bilmediğimiz bir yere ilk defa gidilecek olunması gibi çözüm aranması gereken durumlar günlük hayatta karşılaşılabilecek problemlerdir (Görsel 1.1).

1.1.2. Problem Çözme Süreci

Sabah okula gitmek için bindiğiniz otobüsün arıza yapması, teslim tarihi gelmiş olan ödevinizi evde unutmanız veya siz evde değilken gelen kargo görevlisinin sizi evde bulamamış olması bir problem midir? Sizin için bunlar birer problemse nasıl çözülebilir?

Bir problemi çözmesi beklenen alternatif yollar arasından en doğru olanı seçebilmeye problem çözme denir.

Problemi ortaya koyma aşamasından, problemin çözümünün tamamlanmasına kadar geçen zaman ise problem çözme süreci olarak adlandırılır.

Problemleri çözmek için genellikle iki farklı yöntem kullanılır:

1. Deneme yanılma ya da tahminde bulunma yoluyla çözme
2. Algoritma geliştirme yoluyla çözme

Günlük hayatta karşılaşılan bazı problemler öngörülebilir olmadığı gibi bu problemlerin tek bir çözümü de yoktur. Bu gibi problemleri belli adımlar takip ederek çözmek mümkün değildir. Örneğin yemek pişirilirken ne kadar tuz koyulması gerektiği ya da yemeğin ne kadar süre pişirilmesi gerektiği deneme yanılma yoluyla bulunabilir. Çünkü bu soruların cevabı kişiden kişiye değişiklik gösterecektir. Ancak bilgisayarlar aracılığıyla çözmek istediğimiz problemlerin çözüm adımlarının, önceden belirgin bir şekilde ortaya konulması gerekir. Bu amaçla hazırlanan yöntemlere algoritma denir. Algoritmalar bir problemi çözmek veya belirli bir işi yapmak için adım adım tanımlanmış işlemler kümesidir. Problem çözme sürecini kolaylaştırmak ve hata yapma ihtimalini azaltmak için öncelikle mevcut durumun problem olup olmadığına karar verilmeli ve problem net olarak ortaya konulmalıdır.

¹Papirüs, eski Mısırlıların bu bitkinin saplarından yaptıkları kâğıt ("<https://sozluk.gov.tr>").



Görsel 1.1: Problem

Problem net bir şekilde ortaya konulup iyi analiz edildikten sonra çözüm için uygun planlamalar yapılmalıdır. Daha sonra çözüm için farklı yollar düşünülüp bu çözüm yolları arasından uygun olabilecek çözüm yolu seçilmelidir. Problem çözüldükten sonra kontrol edilmelidir.



Şekil 1.1: Problem çözme süreci

Her bir aşamada yapılması gereken işlemler aşağıda açıklanmıştır.

1. Problemi tanımlama: Problemin ne olduğu belirgin bir şekilde ortaya konulmalıdır.
2. Problemi anlama: Problemin kaynağının ne olduğu ve problem çözüldükten sonra beklenen faydalar belirlenmelidir. Bir problem ne kadar iyi anlaşılırsa çözümü o kadar kolay olacaktır.
3. Alternatif çözüm yollarını belirleme: Problemi çözmesi beklenen tüm alternatifler sıralanmalıdır.
4. En uygun çözümü seçme: Bir önceki adımda belirlenen alternatifler arasından en uygun olanının seçilmesi gerekir. Bunun için alternatiflerin artıları ve eksileri yazılabilir. Çoğu zaman çözüme en hızlı ulaştıran alternatif doğru çözüm olarak görünse de bu durum her zaman geçerli değildir. En hızlı çözümün güvenli olmadığı, maliyet açısından kabul edilebilir olmadığı ya da uzun ömürlü olmadığı durumlar ortaya çıkabilir. Böyle durumlarda tüm faktörler göz önünde bulundurularak en uygun çözüm seçilmelidir.
5. Çözümü uygulama: Bir önceki adımda belirlenen çözüm yöntemi kullanılarak problemi çözme işi gerçekleştirilir. Çözüm adımlarının kafa karışıklığına yol açmayacak bir şekilde ortaya konulması gerekir.
6. Çözümü test etme: Uygulanan çözümün beklentileri yerine getirip getirmediği test edilmelidir. Uygulanan çözümün hataları varsa bunları gidermek için önceki işlem basamaklarına dönülmesi gerekebilir.

Örnek 1: Sabahları okula geç kalma problemini, problem çözme adımlarını kullanarak çözmeye çalışalım.

Problemi tanımlama: Okula geç kalma durumunun bir problem olduğunun farkına varılması problemi tanımlama aşamasıdır.

Problemi anlama: Problemin kaynağının ne olduğu tespit edilmelidir. Geç kalmaya birçok şey neden olabilir. Bunlar; uygun ulaşım alternatifini seçmeme, geç uyumaya bağlı geç uyanma, hazırlanma aşamasının uzun sürmesi, okulun çok uzak olması vs. Akşamları bilgisayarda çok fazla oyun oynamaktan kaynaklı geç yatılabildiği, bu nedenle okula geç kalındığı düşünülerek buna yönelik bir çözüm bulmaya çalışılmalıdır.

Alternatif çözüm yolları belirleme: Bu problemi çözmek için aşağıdaki alternatifler kullanılabilir.

- Bilgisayarı evden çıkarmak
- Ebeveynlerimizden yatma zamanı geldiğinde bizi uyarmasını istemek
- Bilgisayardan oyunları kaldırmak
- Oyun için ayrılacak günlük süre belirleyip buna uymak

En uygun çözümü seçme: Yukarıda belirlenen her bir alternatifin artıları ve eksileri yazılarak en uygun olanı seçilmeye çalışılmalıdır.

Bilgisayarı evden çıkarma: Hızlı bir çözümdür ancak bazı araştırma ve ödevlerin bilgisayar kullanarak yapıldığını ve evdeki diğer bireylerin de bilgisayar kullandığını düşünürsek uygulanabilir değildir.

Ebeveynlerden yatma zamanı geldiğinde bizi uyarmasını isteme: İşe yarayabilecek bir yöntem olmasına rağmen kişisel sorumluluklarımızı başkalarına yüklemiş olacağımızdan ve iç disiplinemizi sağlamada yetersiz kalacağından mantıklı değildir.

Bilgisayardan oyunları kaldırma: En hızlı alternatiftir ancak bir alışkanlığı bir anda bırakmak kolay olmayacağı için uygulanabilir değildir.

Oyun için ayrılacak günlük süre belirleyip buna uyma: En uygun alternatiftir.

Çözümü uygulama: Kendinize bilgisayarda günlük bir saat oyun oynama limiti koyduğunuz ve bir süre bunu uyguladığınızı düşününüz.

Çözümü test etme: Okula geç kalma davranışının azalıp azalmadığına bakılarak çözüm test edilebilir.

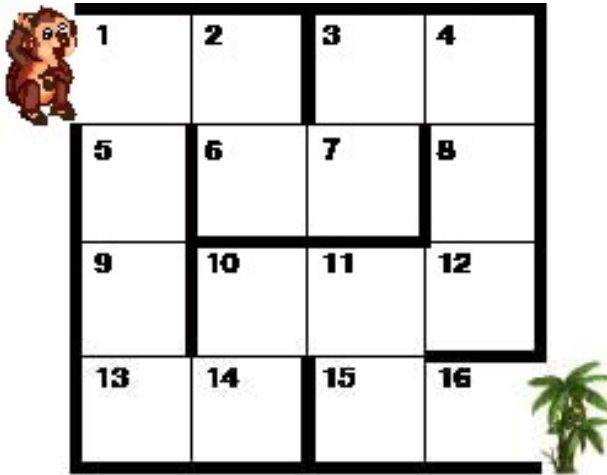
Örnek 2: Bir dağ köyünün tepesinde çiftliği bulunan Hasan Amca'nın Ali ve Ahmet adında iki oğlu varmış. Ali her zaman düzenli ve planlı, Ahmet ise oldukça aceleci birisiymiş. Günlerden bir gün Hasan Amca hem kalıcı hem de insanlara faydalı olsun diye köye bir çeşme yaptırmaya karar vermiş. Çeşme yapımı için gerekli malzemelerin karşığı köyden temin edilmesi gerekiyormuş. Bunun için iki oğlunu yanına çağırıp "Hanginiz malzemeleri alıp işe önce başlasa çiftlik yönetimini ona devredeceğim." demiş. Hep aceleci oluşuyla tanınan Ahmet, alınacaklar listesini hazırlamadan ve güzergâhı belirlemeden yola çıkmış. Bir de karşıdaki köye erken ulaşabilmek için kısa yoldan gitmeye çalışırken yolunu kaybetmiş. Üstüne üstlük malzeme listesini hazırlamadığı için de eksik malzemelerle ve vakit kaybederek çiftliğe geri dönmüş. Ali ise önce iş planını yapmış. Alınacaklar listesini ve yol güzergâhını önceden belirlediği için erkenden malzemeleri alıp gelerek çeşmeyi yapmaya başlamış ve hızlı bir şekilde bitirmiş. Böylelikle de çiftliğin yönetimi babası tarafından ona hediye edilmiş.



Görsel 1.2: Dağ köyü

Bu örnek; problemi çözmeye çalışırken planlamanın ne kadar önemli olduğunu, plansız hareket edildiğinde boşa emek harcanacağını ve sonuca ulaşmanın çok zor olacağını göstermektedir.

Örnek 3: Aşağıdaki görselde 16 hücreli bir labirent verilmiştir. Labirentte bir başlangıç ve bir bitiş noktası vardır. Amaç; sevimli maymunun duvarlardan atlamaadan, herhangi bir hücreye ikinci kez uğramadan ve en kısa yoldan muz ağacına gidebilmesidir.

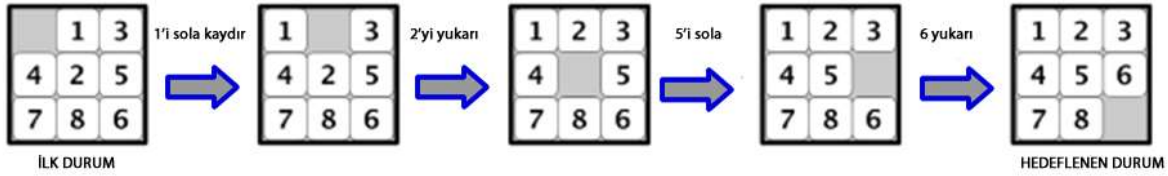


Görsel 1.3: Labirent oyunu örneği

Bulmacalar; yaratıcılığı geliştiren ve analitik düşünmeyi sağlayan, iyi tanımlanmış kurallara ve hedeflere sahip, problem çözme becerilerini geliştiren oyunlardır.

Sevimli maymun, 1-5-9-13-14-10-11-15-16 veya 1-2-6-7-3-4-8-12-11-15-16 adımlarını izleyerek muz ağacına ulaşabilir. Birden fazla çözümü olan örnek problemde en uygun çözüm en kısa çözüm olacaktır. Ancak bir problemi çözerken en kısa çözüm her zaman en doğru çözüm olmayabilir.

Örnek 4: 3*3'lük bir ızgara üzerinde 1'den 8'e kadar numaralandırılmış fayanslar yer almaktadır. Buradaki amaç, mümkün olduğunca az hamle kullanarak fayansları hedefteki görüntüye gelecek şekilde yeniden düzenlemektir.



Görsel 1.4: Problem çözme örneği

İzlenecek adımlarda fayanslar yatay veya dikey olarak boş olan karelere kaydırılır. Yukarıda başlangıç tahtasından hedef tahtasına giderken izlenecek adımlar gösterilmiştir.

1.2. Problem Çözmede Temel İşlemler

Bilgisayarlar, tüm işlemleri matematiksel işlemler yaparak gerçekleştirir. Bu nedenle problem çözmede kullanılacak temel matematiksel işlem ve kavramları bilmek gerekir. Bunlar aritmetiksel ve mantıksal operatörler ile karşılaştırma operatörleridir.

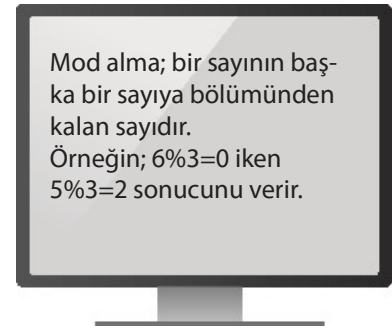
Operatörler önceden tanımlanmış birtakım matematiksel ya da mantıksal işlemleri yapmak için kullanılan özel karakter ya da karakterler topluluğudur (Algan, 2008).

1.2.1. Aritmetiksel Operatörler

Toplama, çıkarma, çarpma, bölme, üs alma ve mod alma gibi matematik işlemlerinin yapıldığı operatörlerdir.

Tablo 1.1: Aritmetiksel Operatörler Tablosu

Operatör Adı	Sembolü	Örnek
Toplama	+	3+6
Çıkarma	-	6-3
Çarpma	*	3*6
Bölme	/	6/3
Üs Alma	**	6**3
Mod alma	%	6%3



1.2.2. Karşılaştırma Operatörleri

Karşılaştırma işlemi yapılması gereken durumlarda kullanılan operatörlerdir.

Tablo 1.2: Karşılaştırma Operatörleri Tablosu

Operatör Adı	Sembolü	Örnek
Eşittir	==	ad=='özge'
Eşit Değildir	!=	ad!='özge'
Büyüktür	>	a>45
Küçüktür	<	a<45
Büyük Eşittir	>=	5>=a
Küçük Eşittir	<=	a<=5

Karşılaştırma operatörleri, karşılaştırma sonunda true (doğru) veya false (yanlış) değeri döndürür.

Örnek 1: $2 > 1$ ifadesi "true" değerini döndürürken, $5 < 2$ ifadesi "false" değerini döndürür.

Örnek 2: $i = 50$ olsun. $i < 30$ ifadesi "false" değerini döndürecek.

Sıra Sizde: $4 == 4$ ifadesi hangi değeri döndürür?

1.2.3. Mantıksal Operatörler

"ve", "veya", "değil" gibi mantıksal işlemleri yapan operatörlerdir.

"Ve" operatörü, iki veya daha fazla koşulun tümünün doğru olduğu durumlarda "doğru" sonucunu veren operatördür. Günlük hayattaki kullanımıyla aynıdır.

Örneğin makarna yapmak için su, tuz ve makarna gerekir. Bunlardan herhangi biri olmadan makarna yapılamaz.

"Veya" operatöründe, iki veya daha fazla koşuldan en az birinin doğru olması durumunda sonuç "doğru" olur. Bu da günlük hayatta kullandığımız gibidir.

Örneğin bir havuzu boşaltmak için kırmızı, yeşil ve mavi renkte üç farklı musluğumuz olsun. Havuzun boşalması için kırmızı musluğu veya yeşil musluğu veya mavi musluğu açmamız yeterli olacaktır. Dilersek iki ya da üç musluğu aynı anda açarak da havuzu boşaltabiliriz.

"Değil" operatörü ise mantıksal bir durumu tersine çevirir. Sonucu "doğru" olan bir mantıksal sınamayı "yanlış"a, sonucu "yanlış" olan bir mantıksal sınamayı ise "doğru"ya çevirir. Günlük hayattaki olumsuzluk ifadelerine karşılık gelir.

Örneğin annelerimiz meyve alırken pazarcıya "Sağlamlarından ver." diyebilir. "Çürüklerinden verme." deseler de pazarcı yine aynı şeyi anlayacaktır. Çünkü bir meyve ya çürük ya da sağlam olur. İkinci ifadede kullanılan olumsuzluk ekinin koşulu tersine çevirdiğine dikkat ediniz.

Tablo 1.3: Mantıksal Operatörler Tablosu

Operatör Adı	Sembolü	Örnek
Ve (and)	and	$a < 4$ and $a > 8$
veya (or)	or	$a < 4$ or $a < 3$
değil (not)	not	not($a == b$)

Mantıksal operatörlerin doğruluk tablosunda gösterilmiş hâli aşağıdaki gibidir.

Tablo 1.4: Mantıksal Operatörlerin Doğruluk Tablosunda Gösterilişi

ve			veya			değil	
a	b	a and b	a	b	a or b	a	a'
1	1	1	1	1	1	1	0
0	1	0	0	1	1	0	1
1	0	0	1	0	1		
0	0	0	0	0	0		

1.2.4. İşlem Önceliği

Aritmetik işlemler yapılırken kullanılan operatörlerde öncelik sıralaması vardır.

Bu sıralama;

Parantez ()

Üs alma **

Çarpma - Bölme *, /

Toplama - Çıkarma +, - şeklindedir.

Örnek 1: $10+4*3/(8+4)=?$ işlemini yapınız.

Çözüm: $=10+4*3/12$

$=10+12/12$

$=10+1$

$=11$ olacaktır.

Örnek 2: $(2^2+4)-8=?$ işlemini yapınız.

Çözüm: $=(4+4)-8$

$=8-8$

$=0$ olacaktır.

Sıra Sizde: $20+16/4-10*1+5=?$ işleminin sonucu nedir?

1.3. Algoritmalar

Algoritma kelimesi bir İslam Bilgini olan El-Harezmi'nin (780-850) 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, 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.

Algoritmalar;

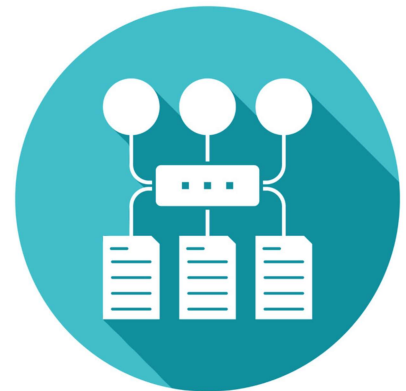
Açık ve net olmalıdır.

Kullanılacak olan girdiler iyi tanımlanmış olmalıdır.

Çıktılar açık ve anlaşılır olmalıdır.

Algoritmalar hızlı olmalıdır.

Sonlu ve uygulanabilir olmalıdır.



Görsel 1.5: Algoritma

Örnek 1: : Evimize gelen arkadaşımıza Türk kahvesi ikram edelim mi?

2 kişilik sade Türk kahvesi hazırlama algoritması aşağıdaki gibi olacaktır:

Adım 1- Başla

Adım 2- Cezveye iki fincanlık su koy

Adım 3- Cezveye 2 tatlı kaşığı kahve koy

Adım 4- Ocağı yak

Adım 5- Cezveyi ocağa koy

Adım 6- Kahveyi karıştır

Adım 7- Bir süre kahvenin olmasını bekle

Adım 8- Kahve köpürmeye başladı mı? Hayır ise 6. Adıma git

Adım 9- Kahveyi fincanlara doldur

Adım 10- Bitir

Örnekte görüldüğü gibi her algoritma bir başlama komutu ile başlar ve bitirme komutu ile sona erer. Problemi çözmek için problem küçük parçalara bölünür ve hedeflenen sonuca ulaşmak için atılan adımlar net ve uygulanabilir niteliktedir.

Örnek 2: Öğretmeninizin sizi soru çözmek için tahtaya kaldırmış olduğunu düşününüz. Algoritmayı çözmek için işlem basamakları aşağıdaki gibi olacaktır:

Adım 1- Başla

Adım 2- Ayağa kalk

Adım 3- Tahtanın önüne git

Adım 4- Tahta kalemını al

Adım 5- Soruyu çöz

Adım 6- Soruyu doğru çözdün mü? Evet ise 8. Adıma git

Adım 7- Tahtayı sil ve 5. Adıma git

Adım 8- Yerine otur

Adım 9- Bitir

Sıra Sizde: Sizler de çay demleme algoritmasını metinsel olarak hazırlayınız.

1.3.1. Sözde Kod (Pseudo-code)

Konuşma dili ile programlama dili arasında, algoritma geliştirmek için kullanılan yapay kodlara sözde kod denir. Sözde kodlar, günlük dille ifade edilmiş (metinsel olarak tanımlanmış) problemlerin programlamaya yaklaştırılmış hâlidir.

Örnek 1: Girilen sayının karesini bulan algoritmayı metinsel ve sözde kod kullanarak yazınız.

Değişken: Her seferinde farklı değerler alabilen ifadelerdir.

Sabit: Değeri değişmeyen ifadelerdir.

Metinsel algoritma	Sözde kod
Adım 1- Başla	Adım 1- Başla
Adım 2- Sayıyı oku	Adım 2- Oku a
Adım 3- Sayının karesini hesapla	Adım 3- $kare=a*a$
Adım 4- Sonucu ekrana yaz	Adım 4- Yaz kare
Adım 5- Bitir	Adım 5- Bitir



Bu örnekte "a" ile "kare" birer değişkendir.

Örnek 2: İki sayıyı çarpıp sonucu ekrana yazdıran algoritmayı metinsel ve sözde kod kullanarak yazınız.

Metinsel algoritma	Sözde kod
Adım 1- Başla	Adım 1- Başla
Adım 2- Birinci sayıyı oku	Adım 2- sayi1'i oku
Adım 3- İkinci sayıyı oku	Adım 3- sayi2'yi oku
Adım 4- İki sayıyı çarp	Adım 4- $carpim=sayi1*sayi2$
Adım 5- Çarpımı ekrana yaz	Adım 5- Yaz carpim
Adım 6- Bitir	Adım 6- Bitir



Bu örnekte sayi1, sayi2 ve carpim birer değişkendir.

Örnek 3: Bir öğrencinin matematik dersinden aldığı iki notun ortalamasını hesaplayan algoritmayı metinsel ve sözde kod kullanarak yazınız.

Metinsel algoritma	Sözde kod
Adım 1- Başla	Adım 1- Başla
Adım 2- Birinci notu oku	Adım 2- Oku not1
Adım 3- İkinci notu oku	Adım 3- Oku not2
Adım 4- Ortalamayı hesapla	Adım 4- $ortalama=(not1+not2)/2$
Adım 5- Ortalamayı ekrana yaz	Adım 5- Yaz ortalama
Adım 6- Bitir	Adım 6- Bitir



Sıra Sizde: Dik üçgenin alanını hesaplayan metinsel algoritmayı ve sözde kodu yazınız.

Örnek 4: Klavyeden girilen iki sayının büyük olanından küçük olanını çıkaran algoritmayı yazınız.

Metinsel algoritma

- Adım 1- Başla
- Adım 2- Oku sayi1
- Adım 3- Oku sayi2
- Adım 4- Eğer $sayi1 > sayi2$ ise $sonuc = sayi1 - sayi2$
- Adım 5- Değilse $sonuc = sayi2 - sayi1$
- Adım 6- Yaz sonuc
- Adım 7- Bitir

Mantıksal algoritmalar:
Algoritma içinde karşılaştırma yapma veya karar vermeyi gerektiren durumlar için kullanılır.

Örnek 5: Suyun sıcaklığına göre maddenin katı, sıvı veya gaz olma durumunu gösteren algoritmayı hazırlayınız.

- Adım 1- Başla
- Adım 2- Oku sıcaklik
- Adım 3- Eğer $sicaklik \leq 0$ ise "katı" yaz
- Adım 4- Eğer $sicaklik > 0$ ve $sicaklik < 100$ ise "sıvı" yaz
- Adım 5- Eğer $sicaklik \geq 100$ ise "gaz" yaz
- Adım 6- Bitir

Sıra Sizde: Klavyeden girilen yaş değeri 18'den büyük ve eşitse "Reşittir", aksi hâlde "Reşit değildir" yazan algoritmayı hazırlayınız.

Örnek 6: Ekranı 5 defa "merhaba" yazdıran algoritmayı yazınız.

- Adım 1- Başla
- Adım 2- $sayac = 0$
- Adım 3- yaz "merhaba"
- Adım 4- $sayac = sayac + 1$
- Adım 5- Eğer $sayac < 5$ ise git Adım 3
- Adım 6- Bitir

Döngüsel algoritmalar:
Algoritma içinde tekrar eden işlemler(döngü) için kullanılır.

Örnek 7: 0'dan 100'e kadar olan çift sayıları ekrana yazdıran algoritmayı hazırlayınız.

- Adım 1- Başla
- Adım 2- $sayac = 0$
- Adım 3- Yaz sayac
- Adım 4- $sayac = sayac + 2$
- Adım 5- Eğer $sayac \leq 100$ ise git Adım 3
- Adım 6- Bitir

Sayaç: Bir işlemin belli bir sayıda artması veya azalması şeklindeki sayma işlemlerinde kullanılan değişken.

Sıra Sizde: Klavyeden girilen 5 adet sayıdan 20'den küçük olanların sayısını gösteren algoritmayı hazırlayınız.

1.3.2. Problem Çözmede Algoritma Hataları

Bir problemin çözümüyle ilgili algoritma hazırlayıp işleme aldığımızda hatalı sonuçlar meydana gelebilir. Bu gibi hatalara algoritma hataları denir.

Örnek 1: Evimizin bahçesinde kedimiz ve köpeğimiz olduğunu, bunlara ayrı ayrı mama kapları kullandığımızı düşünelim (Kedinin kabı mavi renkli, köpeğin kabı beyaz renkli olsun.).

Kediyi besleme problemini algoritma hâlinde aşağıdaki şekilde yazalım.

Adım 1- Başla

Adım 2- Kedi mamasını al

Adım 3- Yemek kabına koy

Adım 4- Bitir

Bu algoritmayı adım adım inceleyecek olursak;

Adım 1- Başla (Başla komutu ile işlemleri başlattık.).

Adım 2- Kedi mamasını al (Kedi mamasını aldık.).

Adım 3- Yemek kabına koy (Burada kedi mamasını kaba koyacağız ancak kabın rengi belirtilmediği için hangi kaba mama koyacağımızı bilmiyoruz.).

Karşılaşılan bu gibi hatalara algoritma hataları denilir ve bu hatalar, yazılmış olan kodlar adım adım incelenerek çözülür.

Örnek 2: Ellerimizi yıkama işleminin algoritmasını aşağıdaki şekilde yazalım.

Adım 1- Başla

Adım 2- Musluğu aç

Adım 3- Ellerini yıka

Adım 4- Bitir

Bu algoritmayı adım adım inceleyecek olursak;

Adım 1- Başla (Başla komutu ile işlemleri başlattık.).

Adım 2- Musluğu aç (Musluk açıldı.).

Adım 3- Ellerini yıka (Burada eller yıkıyor ancak önemli bir mantık hatası var. Çünkü ellerimizi sabunla yıkamamız gerekir. Ayrıca işlemi bitirmeden önce musluğu kapatmamız gerekir.).

Algoritmayı düzenleyerek yeniden yazacak olursak;

Adım 1- Başla

Adım 2- Musluğu aç

Adım 3- Ellere sabun al

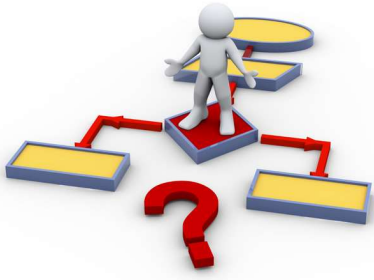
Adım 4- Ellerini yıka

Adım 5- Musluğu kapat

Adım 6- Bitir

şeklinde olmalıdır.

Sıra Sizde: Sınıfınızda üçer kişilik gruplar oluşturunuz. Herkes ayrı ayrı ATM'den para çekme algoritmasını yazdıktan sonra her grupta algoritmaları birlikte inceleyerek hatalarını gidermeye çalışınız.



Görsel 1.6: Akış diyagramı

1.4. Akış Diyagramları

Algoritma ile adım adım yapılan işlemlerin, özel semboller (geometrik şekiller) kullanılarak gösterilmesine akış diyagramı denir.

Akış diyagramı hazırlanırken kullanılan şekiller ve anlamları Tablo 1.5'te gösterilmiştir.

Tablo 1.5: Akış Diyagramı Sembol ve Görevleri

ŞEKİLLER	KULLANIM YERİ
	Başla-Bitir Algoritmanın başladığını ve bittiğini gösteren semboldür.
	Aritmetik işlemlerin ve değişkene değer atama işlemlerinin yapıldığı semboldür.
	Veri giriş ve değişken tanımlamalarının yapıldığı semboldür.
	Önceden tanımlı işlem / fonksiyonları çalıştırmak için kullanılan semboldür.
	Çıktı almak ve ekran görüntüsü oluşturmak için kullanılan semboldür.
	Döngü işlemleri için kullanılan semboldür.
	Karar verme / karşılaştırma işlemleri için kullanılan semboldür.
	Akış noktalarını bağlamak için kullanılan semboldür.
	Akış yönünü gösterir. Akış, okun yönüne göre ilerlemektedir.

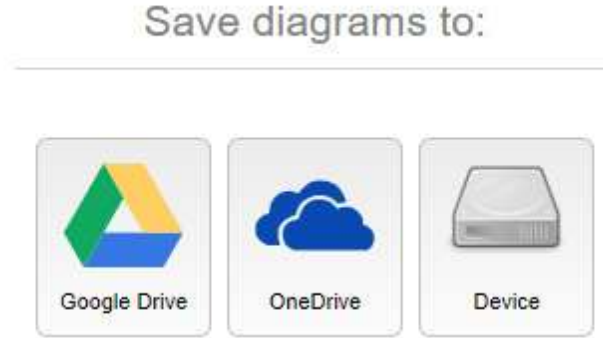
1.4.1. Flowchart (Akış Diyagramı) Hazırlama Programının Kurulumu

Akış diyagramı hazırlama programı kullanılarak hazırlanmak istenilen akış şemaları bilgisayar ortamında kolaylıkla çizilebilir.

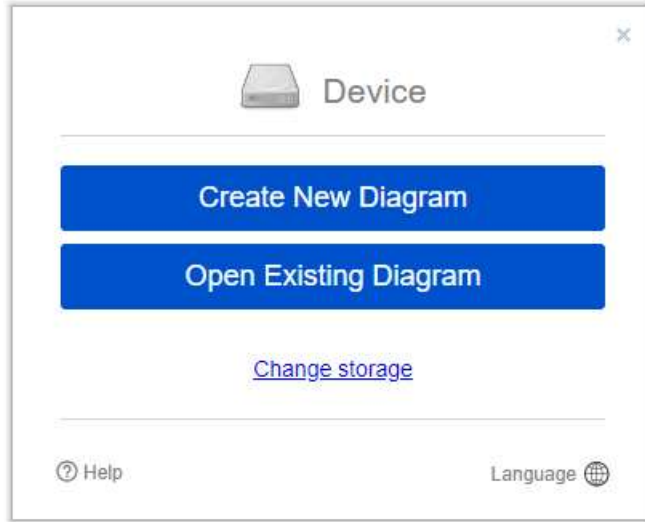
Akış diyagramı hazırlama programı bilgisayara kurulum gerektirmeden çevrimiçi kullanılabilceği gibi bilgisayara kurularak internet bağlantısı olmadan da kullanılabilir.

Akış diyagramı hazırlama programına draw.io internet adresinden erişilebilir.

Akış diyagramı hazırlama programı tarayıcıda çalıştırıldığı zaman aşağıdaki görüntü ile karşılaşılır:



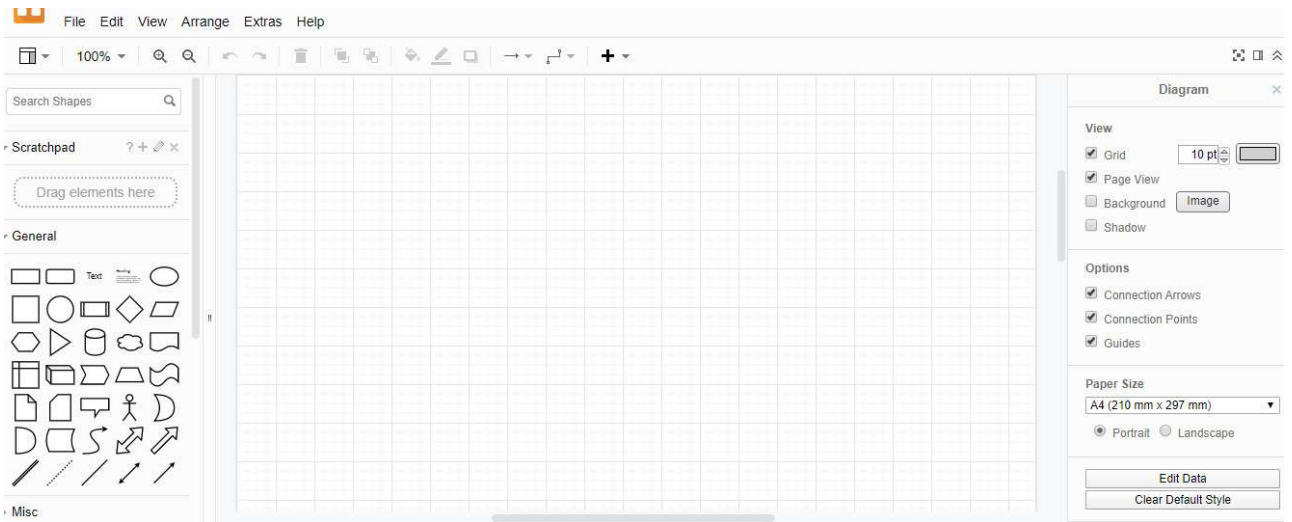
Akış diyagramı programı ile çalışılacak ortam buradan seçilir.



Görsel 1.7: Flowchart programı kurulum

"Get desktop" seçeneği programın .exe dosyasını bilgisayara indirerek internet bağlantısı olmadan da programın kullanılmasını sağlar.

Sırası ile Create New Diagram ardından da Blank Diagram seçilip Create düğmesine tıklandığı zaman akış şemalarının hazırlanacağı arayüz ekranı açılır (Görsel 1.8).



Görsel 1.8: Akış şeması arayüzü

Kullanılacak olan nesnelere sürükleyip bırak yöntemiyle tasarım alanına aktararak akış şemalarını basit bir şekilde oluşturulur.

1.4.2. Doğrusal Akış Şeması Örnekleri



<http://kitap.eba.gov.tr/KodSor.php?KOD=22313>

Örnek 1: Dikdörtgenin alanını hesaplayan algoritmayı ve akış şemasını hazırlayınız.

ALGORİTMA	AKIŞ ŞEMASI
<p>A1-Başla</p> <p>A2-Oku kenar1</p> <p>A3-Oku kenar2</p> <p>A4-$alan=kenar1*kenar2$</p> <p>A5-Yaz alan</p> <p>A6-Bitir</p>	<pre> graph TD Start([Başla]) --> Input[/kenar1, kenar2/] Input --> Process[alan=kenar1*kenar2] Process --> Output[/alan/] Output --> End([Bitir]) </pre>

Örnek 2: Klavyeden girilecek 3 sayıyı toplayıp sonucu ekrana yazdıran programın algoritmasını ve akış şemasını hazırlayınız.

ALGORİTMA	AKIŞ ŞEMASI
<p>A1-Başla</p> <p>A2-Sayıları giriniz; A,B,C</p> <p>A3-Toplam=A+B+C</p> <p>A4-Yaz Toplam</p> <p>A5-Bitir</p>	<pre> graph TD Start([Başla]) --> Input[/A,B,C/] Input --> Process[Toplam=A+B+C] Process --> Output[/Yaz Toplam/] Output --> End([Bitir]) </pre>

Sıra Sizde: Dikdörtgenin çevresini hesaplayan algoritmayı ve akış şemasını hazırlayınız.



<http://kitap.eba.gov.tr/KodSor.php?KOD=22319>

1.4.3. Karar İfadeleri Kullanılarak Hazırlanan Akış Şeması Örnekleri

Örnek 1: Klavyeden girilen bir sayının negatif mi, pozitif mi yoksa sıfır mı olduğunu yazdıran programın algoritmasını ve akış şemasını hazırlayınız.

ALGORİTMA	AKIŞ ŞEMASI
<p>A1- Başla</p> <p>A2- Sayıyı gir;Sayı</p> <p>A3- Eğer Sayı<0 ise yaz "girilen sayı negatiftir" ve A6 ya git</p> <p>A4- Eğer Sayı>0 ise yaz "girilen sayı pozitifdir" ve A6 ya git</p> <p>A5- Girilen sayı sıfırdır</p> <p>A6- Bitir</p>	<pre> graph TD Start([Başla]) --> Input[/Sayı/] Input --> Dec1{Sayı<0} Dec1 --> Out1[/Girilen sayı negatiftir/] Dec1 --> Dec2{Sayı>0} Dec2 --> Out2[/Girilen sayı pozitifdir/] Dec2 --> Dec3{Girilen sayı sıfırdır} Dec3 --> End([Bitir]) Out1 --> End Out2 --> End </pre>

Örnek 2: Klavyeden girilen iki sayıdan birincisi büyük ise toplama, ikincisi büyük ise çarpma işlemi yapan algoritmayı ve akış şemasını hazırlayınız.

ALGORİTMA	AKIŞ ŞEMASI
<p>A1-Başla</p> <p>A2-Sayıları gir; x,y</p> <p>A3-Eğer $x > y$ ise $sonuc = x + y$ ve git A5</p> <p>A4-Değilse $sonuc = x * y$</p> <p>A5-Yaz sonuc</p> <p>A6-Bitir</p>	<pre> graph TD Start([Başla]) --> Input[/x,y/] Input --> Decision{x > y} Decision -- Evet --> Process1[sonuc = x + y] Decision -- Hayır --> Process2[sonuc = x * y] Process1 --> Output[/sonuc/] Process2 --> Output Output --> End([Bitir]) </pre>

Sıra Sizde: Klavyeden girilen iki sayıdan büyük olanı ekrana yazdıran algoritmayı ve akış şemasını çiziniz.

1.4.4. Döngüler Kullanılarak Hazırlanan Akış Şeması Örnekleri

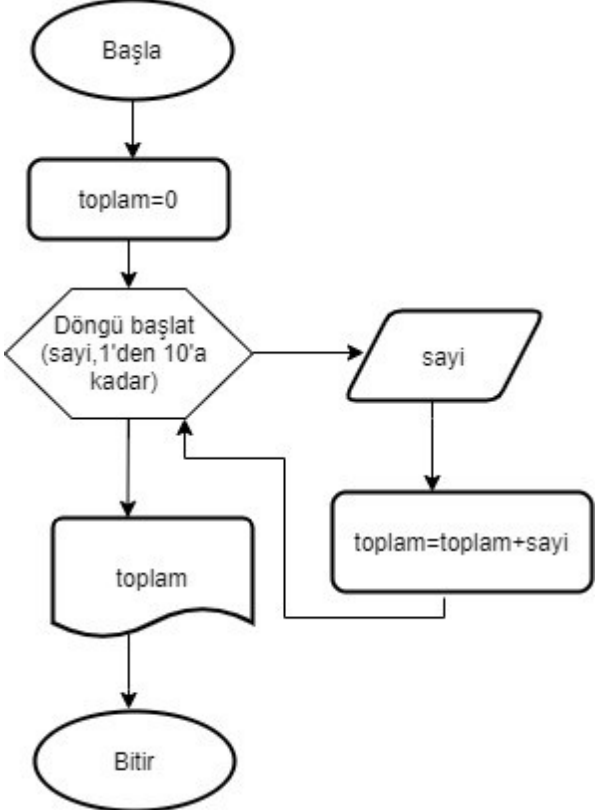


<http://kitap.eba.gov.tr/Kod-Sor.php?KOD=22320>

Örnek 1: 20 Öğrencinin Programlama Temelleri dersi birinci sınav notları giriliyor. Geçme notu 60 olan sistemde, kalan öğrenci sayısını bulan algoritmayı ve akış şemasını hazırlayınız.

ALGORİTMA	AKIŞ ŞEMASI
<p>A1- Başla</p> <p>A2- sayac=0, kalan=0</p> <p>A3- Sınav Notunu Gir, s_not</p> <p>A4- Eğer s_not<60 ise kalan = kalan + 1</p> <p>A5- sayac = sayac + 1</p> <p>A6- Eğer sayac <= 20 ise, Git A3</p> <p>A7- Yaz kalan</p> <p>A8- Bitir</p>	<pre> graph TD Start([Başla]) --> Init[sayac=0 kalan=0] Init --> Input[/s_not/] Input --> Cond1{s_not < 60} Cond1 -- Evet --> IncKalan[kalan=kalan+1] IncKalan --> Cond1 Cond1 -- Hayır --> IncSayac[sayac=sayac+1] IncSayac --> Cond2{sayac < 20} Cond2 -- Evet --> Input Cond2 -- Hayır --> Output[/kalan/] Output --> End([Bitir]) </pre>

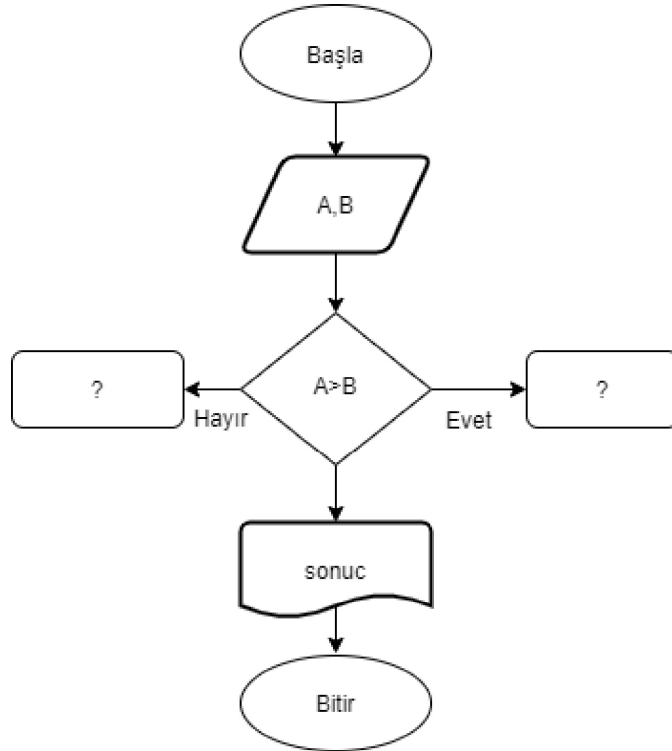
Örnek 2: Klavyeden girilen 10 sayının toplamını hesaplayan algoritmayı ve akış diyagramını hazırlayınız.

ALGORİTMA	AKIŞ ŞEMASI
<p>A1- Başla</p> <p>A2- toplam=0,</p> <p>A3- Döngü başlat (sayi,1'den 10'a kadar)</p> <p>A4- toplam=toplam+sayi</p> <p>A5- Döngüyü bitir</p> <p>A6- Yaz toplam</p> <p>A7- Bitir</p>	 <pre> graph TD Start([Başla]) --> Init[toplam=0] Init --> LoopStart{{Döngü başlat (sayi,1'den 10'a kadar)}} LoopStart --> Input[/sayi/] Input --> Add[toplam=toplam+sayi] Add --> LoopStart Add --> Output[/toplam/] Output --> End([Bitir]) </pre>

Sıra Sizde: Sıfır girilinceye kadar girilmiş olan sayıların karesini hesaplayan algoritmayı ve akış şemasını hazırlayınız.

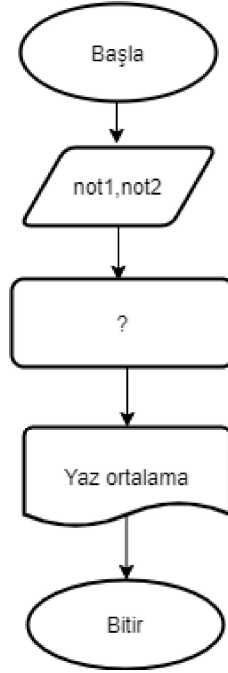
ÖLÇME VE DEĞERLENDİRME 1

1. Aşağıdakilerden hangisi bir problemin çözümü için doğru değildir?
 - A) Bir problemi çözebilmek için problemin net bir şekilde ortaya konması gerekir.
 - B) Problem ne kadar iyi anlaşılırsa çözüm de aynı ölçüde kolay olacaktır.
 - C) Problemi çözmek için farklı yol ve yöntemler denenmelidir.
 - D) Çözüm adımları kafa karıştırmayacak şekilde olmalıdır.
 - E) Çözüme ulaşabilmek için her zaman tek bir yol yeterlidir.
2. Sürekli kilo alma probleminin sebebinin ne olduğunu tespit etme problem çözme basamaklarının hangisine aittir?
 - A) Problemi tanımlama
 - B) Problemi anlama
 - C) Alternatif çözüm yolu belirleme
 - D) Çözümü uygulama
 - E) Çözümü test etme



3. Yukarıdaki tabloda klavyeden girilen iki sayının büyük olanından küçük olanını çıkarıp sonucu ekrana yazdıran programın akış şeması verilmiştir. Programın doğru sonuç vermesi için soru işareti olan yerlere aşağıdakilerden hangisi getirilmelidir?
 - A) Evet: sonuc=A+1, Hayır: sonuc=A-B
 - B) Evet: sonuc=A-B, Hayır: sonuc=B-A
 - C) Evet: sonuc=B-A, Hayır: sonuc=A-B
 - D) Evet: sonuc=sonuc-A, Hayır: sonuc=sonuc-B
 - E) Hiçbiri

4. Klavyeden girilen iki notun ortalamasının bulunması istenmektedir. Algoritma aşağıdaki gibidir.



Buna göre soru işareti olan yere aşağıdakilerden hangisi gelmelidir?

- A) $ortalama=not1+not2$
B) $ortalama=(not1)+(not2)$
C) $ortalama=(not1+not2/2)$
D) $ortalama=(not1+not2)/2$
E) $ortalama=toplam-(not1+not2)$
5. Aşağıda karışık hâlde verilmiş adımları doğru şekilde düzenleyerek "makarna pişirme" algoritmasını hazırlayınız.

Bitir, suyu kaynat, tencereye su koy, başla, makarnayı ekle, tuz ekle, makarnayı pişir, makarnanın suyunu süz.

Adım 1:.....

Adım 2:.....

Adım 3:.....

Adım 4:.....

Adım 5:.....

Adım 6:.....

Adım 7:.....

Adım 8:.....

NOT: Cevaplarınızı cevap anahtarıyla karşılaştırınız. Yanlış cevap verdiğiniz ya da cevap verirken tereddüt ettiğiniz sorularla ilgili konuları veya faaliyetleri geri dönerek tekrarlayınız. Cevaplarınızın tümü doğru ise bir sonraki öğrenme birimine geçiniz.