



2023-2024 Eğitim-Öğretim Yılı
Başiskele Gübretaş Ortaokulu
Bilişim Teknolojileri ve Yazılım Dersi
5.Sınıf

Algoritma ve Programlama

Ders İeriđi:

- Problem özme(yazılım, problem).
- Algoritma kavramı.
- Bir problem özümünde algoritmayı kullanma.
- Akıř řeması bileřenleri ve iřlevleri.
- Akıř řeması çizme.
- Scratch arayüzü ve kod blokları tanıtımı.
- Programlama ile ilgili temel kavramlar.
 - a) Döngü yapısı
 - b) Karakteri Hareket Ettirme, başlatıcılar
 - c) Karakterin Yönü
 - d) Sahnede konum
 - e) Karakteri yön tuřları ile hareket ettirme
 - f) Karar-Mantık yapısı
 - g) Deđişkenler
 - h) Sayaçlar
- Projeler

YAZILIM: Çeşitli görevleri gerçekleştirmek amacıyla hazırlanmış programlara yazılım adı verilir. Her yazılım bir **problemi** çözmek amacıyla geliştirilmiştir.

➤ **PROBLEM:** çözümlmesi gereken sorun ya da aşılması gereken engel anlamına gelir.

➤ Bir Problemin Çözümü İçin:

- Problemi iyi anlamak
- Sonucun doğruluğunu kontrol etmek.
- Kısa ve anlaşılır şekilde çözmek.

➤ Günlük yaşamda karşılaştığımız problemleri bilerek veya farkında olmadan **adım adım** çözmeye çalışırız.

➤ Örneğin yazı yazarken kaleminizin ucu kırıldığında şu adımları takip ederek bu sorunu çözersiniz.

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



Kurt, Kuzu, Ot Problemi



- Tekneye her defasında sadece 1 tanesi alınabiliyor.
- Yani ya kurdu, ya kuzuyu yada otu almanız gerekiyor.
- Eğer kurt ile kuzuyu baş başa bırakırsanız, kurt kuzuyu yer.
- Eğer kuzu ile otu baş başa bırakırsanız, kuzu otu yer.

Çözümü yapalım: <https://www.rekoroyun.com/kurt-kuzu-ot.html>

Kurt, Kuzu, Ot Problemi Çözümü



- ▶ 1) Kuzuyu karşıya geçir.
- ▶ 2) Geri dön.
- ▶ 3) Otu karşıya geçir.
- ▶ 4) Kuzuyu geri getir.
- ▶ 5)Kurdu karşıya geçir.
- ▶ 6) Geri dön.
- ▶ 7)Kuzuyu karşıya geçir.

Çözümü test edelim: <https://www.rekoroyun.com/kurt-kuzu-ot.html>

Bilgisayarlar da problemleri tıpkı bizler gibi adım adım çözmeye çalışır.

Kullandığımız yazılımların tamamı «**kod**» adı verilen bilgisayarın anlayacağı dilde yazılmış özel komutlardan oluşur.

Bu kodlar bilgisayar yazılımcıları tarafından yazılır.

```
31 self.file = None
32 self.fingerprints = set()
33 self.logdups = True
34 self.debug = debug
35 self.logger = logging.getLogger(__name__)
36 if path:
37     self.file = open(os.path.join(path, 'requests.txt'),
38                     'a')
39     self.file.seek(0)
40     self.fingerprints.update(self.request)
41
42 @classmethod
43 def from_settings(cls, settings):
44     debug = settings.getbool('debug', True)
45     return cls(job_dir(settings), debug)
46
47 def request_seen(self, request):
48     fp = self.request_fingerprint(request)
49     if fp in self.fingerprints:
50         return True
51     self.fingerprints.add(fp)
52     if self.file:
53         self.file.write(fp + os.linesep)
54
55 def request_fingerprint(self, request):
56     return request_fingerprint(request)
```



Kodlamaya başlamadan önce oluşturacağımız yazılımın adım adım ne yapacağını tasarlamamız gerekir.

İşte açık ve net ifadelerle problemin adım adım çözümünü gösteren bu taslağa «**algoritma**» adı verilir.

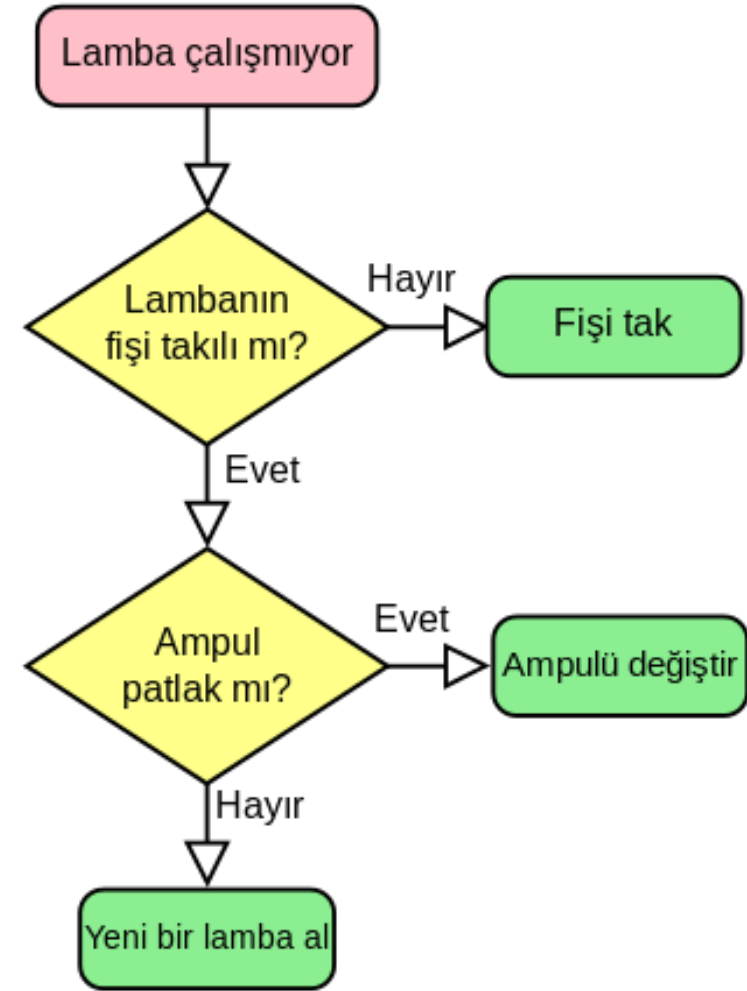
Programlamanın ilk adımı algoritma oluşturmaktır.

► **ALGORİTMA:** Bir problemin çözümünde izlenecek yol anlamına gelir ve problemin çözümünün adımlar halinde yazılmasıyla oluşturulur.





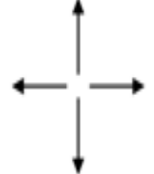


► Algoritma basamaklarının bir başlangıcı ve sonu bulunur.

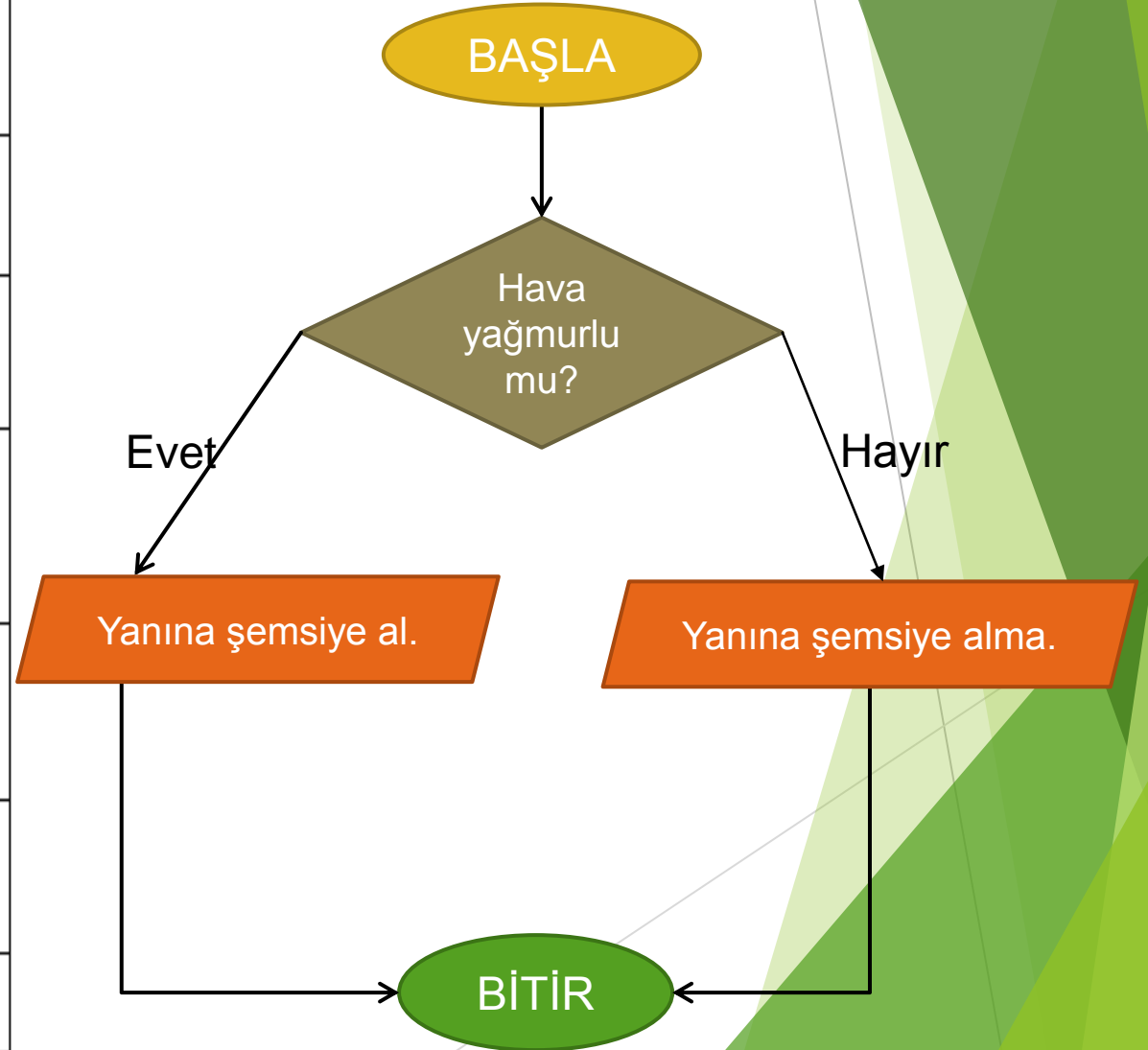
► Her adımda yapılacak işlem açıkça belirtilir.

► Oluşturacağımız yazılımın kusursuz olması için öncelikle her adımını gösteren planını, yani algoritmasını hazırlamalıyız.



AKIŞ DİYAGRAM ŞEKİLLERİ

	Programın başlangıç ve bitiş için kullanılır.
	Bilgi giriş çıkışı için kullanılır.
	Aktarma, aritmetik hesaplama, işlem
	Karar alma için kullanılır.
	Birleştirme çizgileri
	Yazdır
	Bağlantı



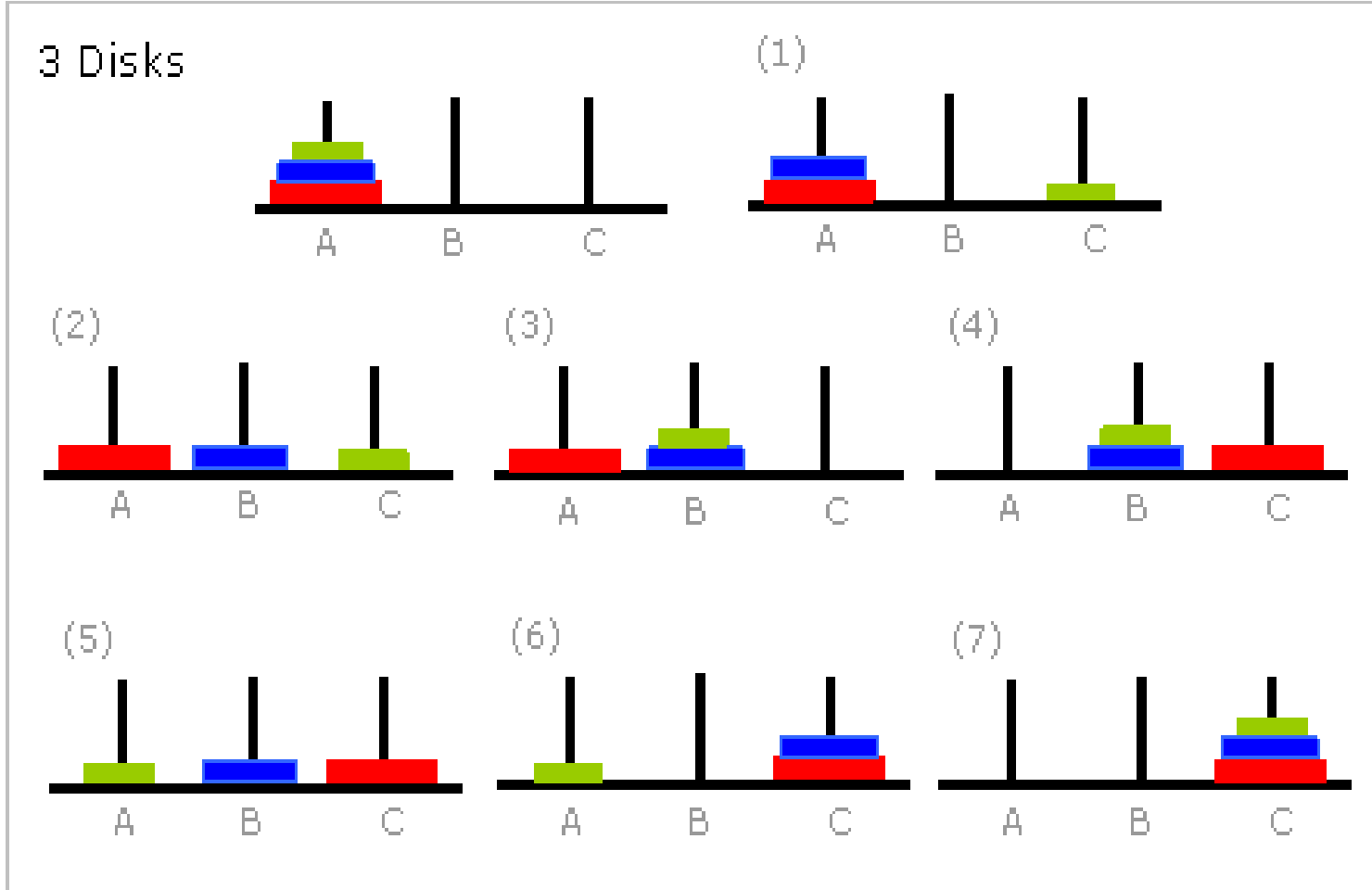
Hanoi Kuleleri



Problemi çözelim: <https://www.rekoroyun.com/hanoi-kuleleri.html>

- Hanoi kuleleri oyununda amaç; En sol taraftaki kulede bulunan diskleri, en sağ tarafta bulunan kuleye büyükten küçüğe doğru olacak şekilde yerleştirmektir.
- Küçük disklerin üstüne kendinden büyük disk koyamazsınız.
- Oyun en az 3 en fazla 6 disk ile oynanmaktadır. Disk sayısı arttıkça oyun zorlaşmaktadır.
- Disk sayısı arttıkça minimum hamle sayınızda buna göre değişmektedir. *Ör: 3 diskte sizden 7 hamlede yapmanız istenirken, 6 diskte ise 63 hamlede tamamlamanız istenmektedir.*
- Oyunda geri alma yoktur. Bu yüzden hamlelerinizi dikkatli yapınız.
- Her hamlede sadece 1 parçayı taşıyabilirsiniz.

Hanoi Kuleleri Çözüm



- 1) Yeşili C'ye götür.
- 2) Maviyi B'ye götür.
- 3) Yeşili B'ye götür.
- 4) Kırmızıyı C'ye götür.
- 5) Yeşili A'ya götür.
- 6) Maviyi C'ye götür.
- 7) Yeşili C'ye götür.

Algoritmayı test edelim: <https://www.rekoroyun.com/hanoi-kuleleri.html>

Ödev: Hanoi kuleleri oyununun 4 diskli olan bölümünü çözerek problemin algoritmasını liste halinde yazmak.



- 1)
- 2)
- 3)
- 4)
- 5)
- 6)
- 7)
- 8)
- 9)
- 10)
- 11)
- 12)
- 13)
- 14)
- 15)

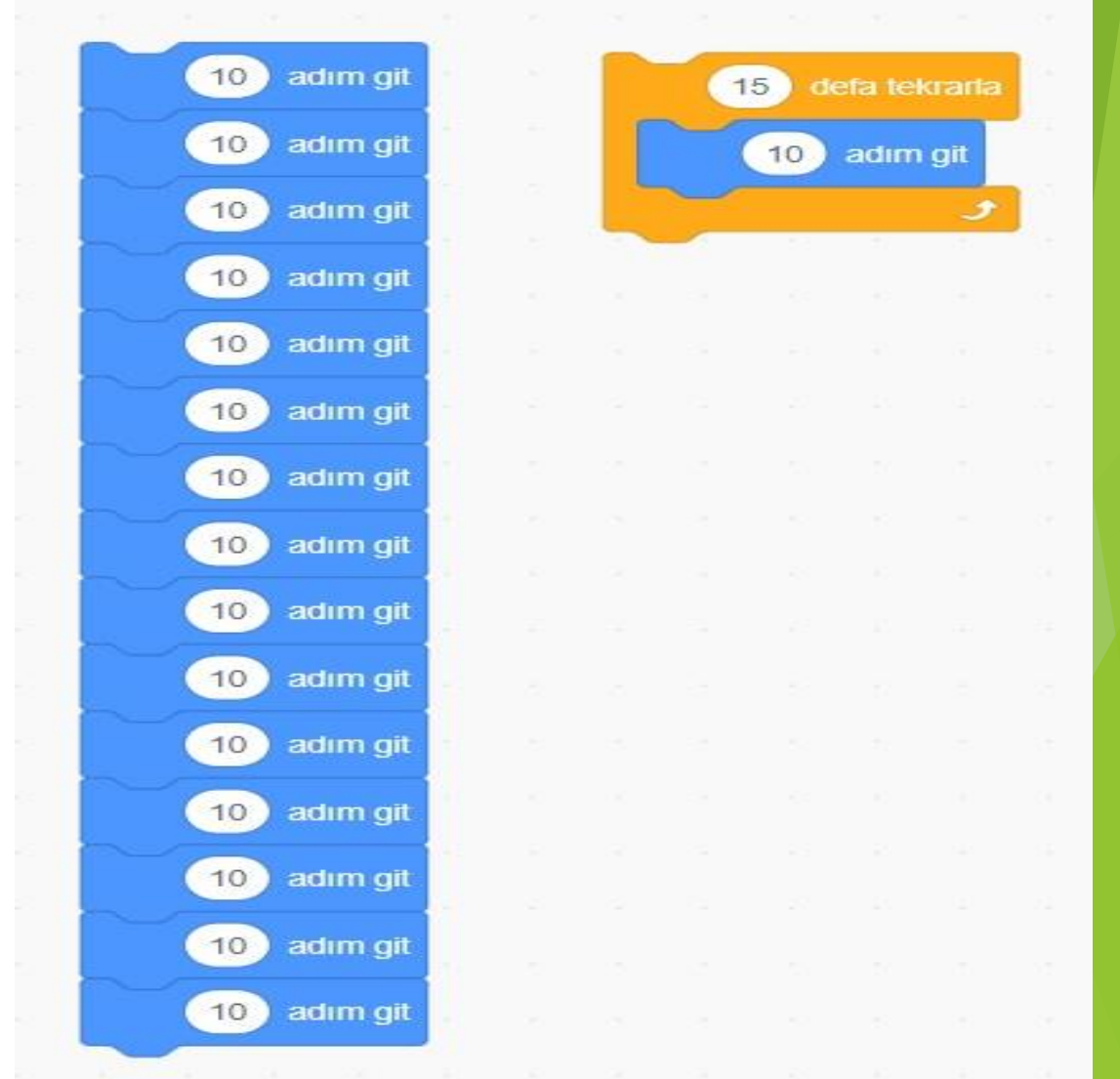
Scratch Arayüz ve Bloklar

Aşağıdaki bağlantıdan tüm kodlara ulaşabilirsiniz.

<https://blockodlama.com/scratch-3-kod-bloklari-tanitimi/>

PROGRAMLAMANNIN TEMELLERİ

- **a) Döngü:** Bir veya birden fazla işlemi, bir koşula bağlı olarak, belirli sayıda veya bir koşul sağlandığı sürece tekrarlayarak çalıştıran kalıplara **döngü** adı verilir.



DÖNGÜ TÜRLERİ



Yazılan sayı kadar içerisinde ki blokları tekrarlatır.

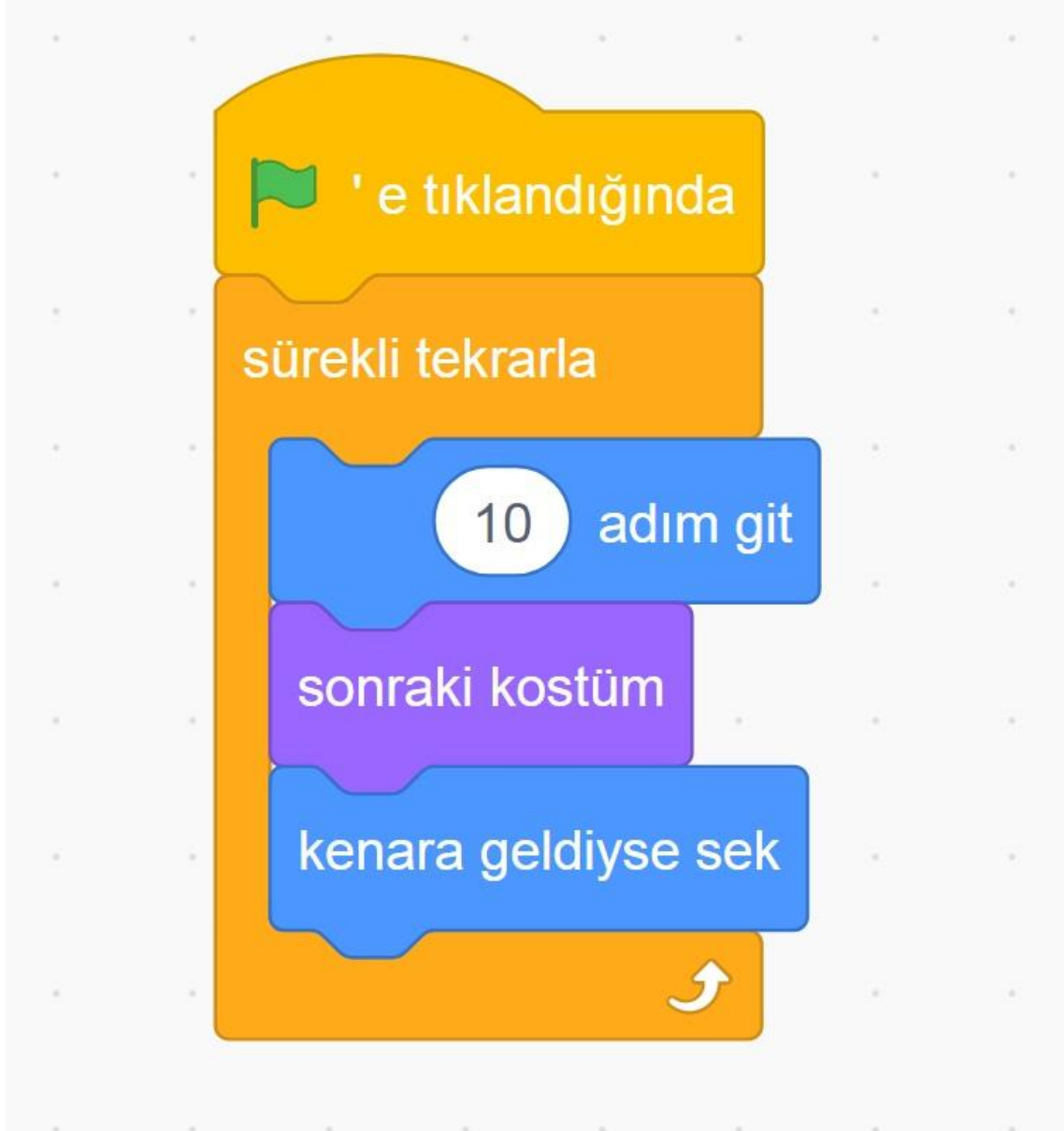


Program durdurulana kadar içerisinde ki blokları tekrarlatır.



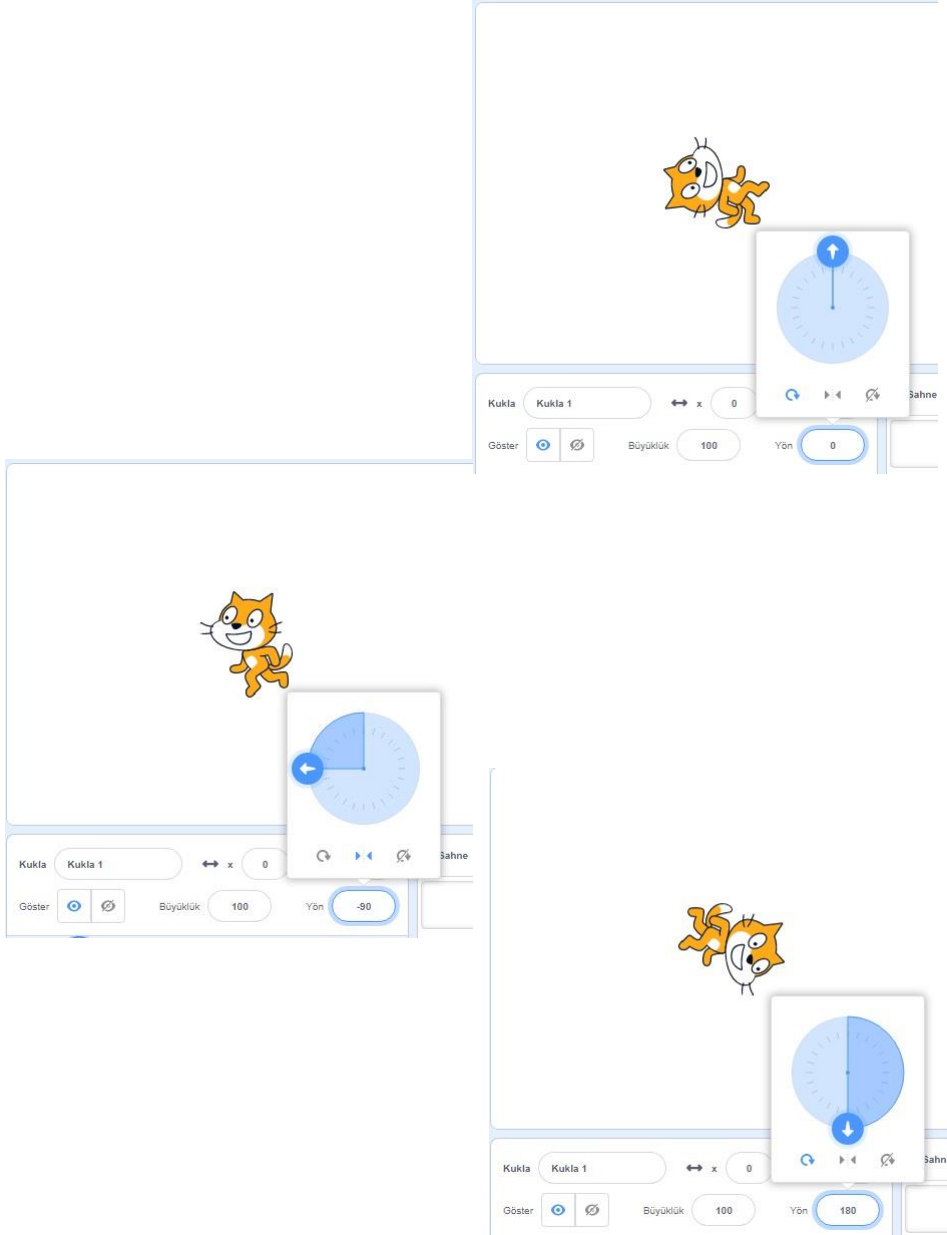
Yazılan koşul gerçekleşene kadar içerisinde ki blokları tekrarlatır. Koşul gerçekleştiği anda tekrarlatmayı durdurur.

b) Karakteri Hareket Ettirme, Başlatıcılar



Yeşil bayrak tıklandığında oyun başlar.
Program durdurulana kadar sürekli tekrarla bloğu çalışır.
Her çalıştığında sonraki kostüme gelir.
Karakter sahnenin köşesine geldiğinde geri döner.

c) Karakterin Yönü



Karakterimizin sahnede hangi yöne bakacağı 0 - 180 derece ve 0 - -179 derece arasında belirlenir.

0 derece ekrana göre yukarı bakar.

90 derece ekrana göre sağa bakar.

180 derece ekrana göre aşağı bakar.

-90 derece ekrana göre sola bakar.

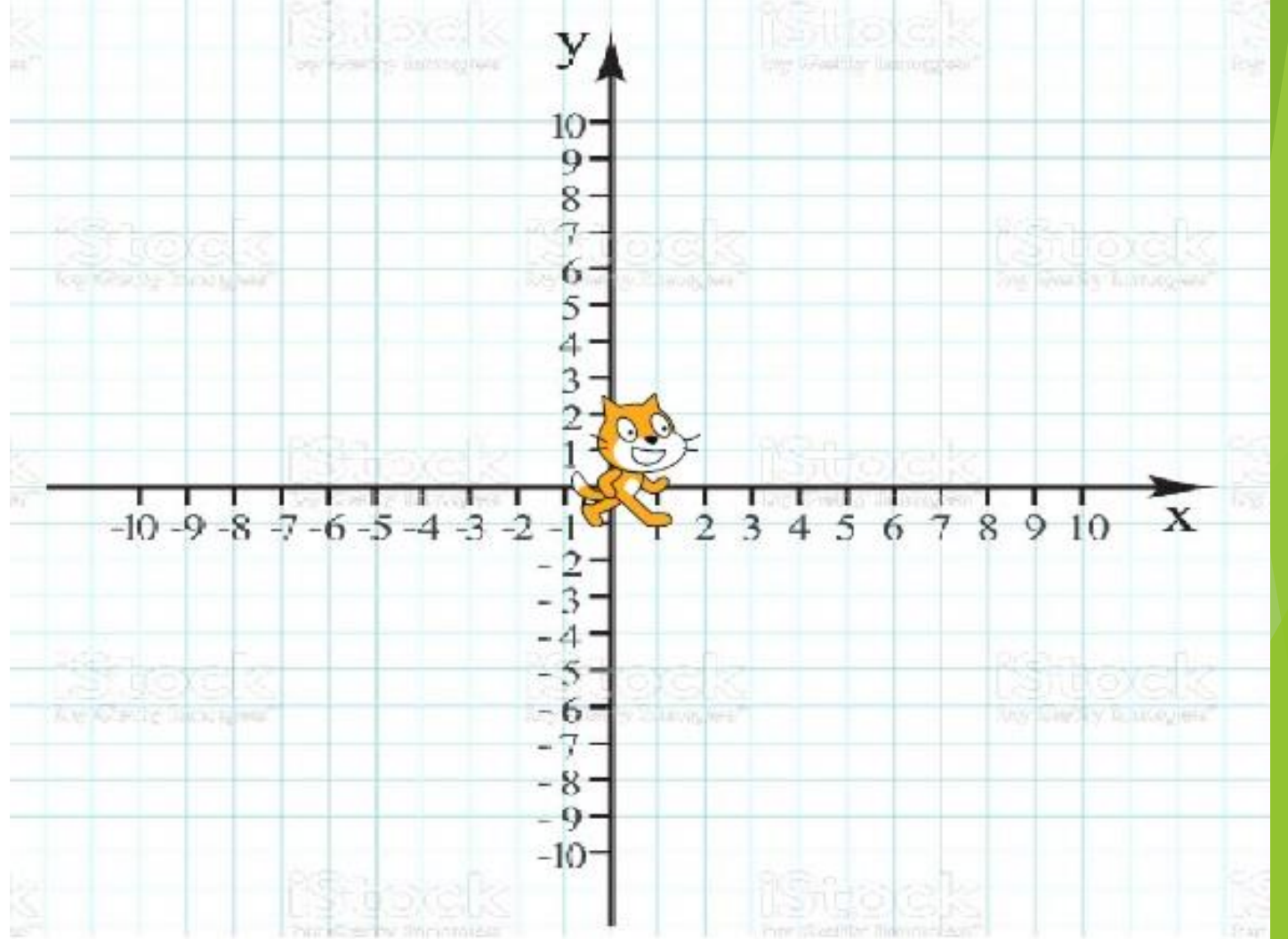
d) Sahnede Konum

Sahnede yer alan karakterimizin konumu önce x eksenine, sonra y eksenine göre belirlenir.

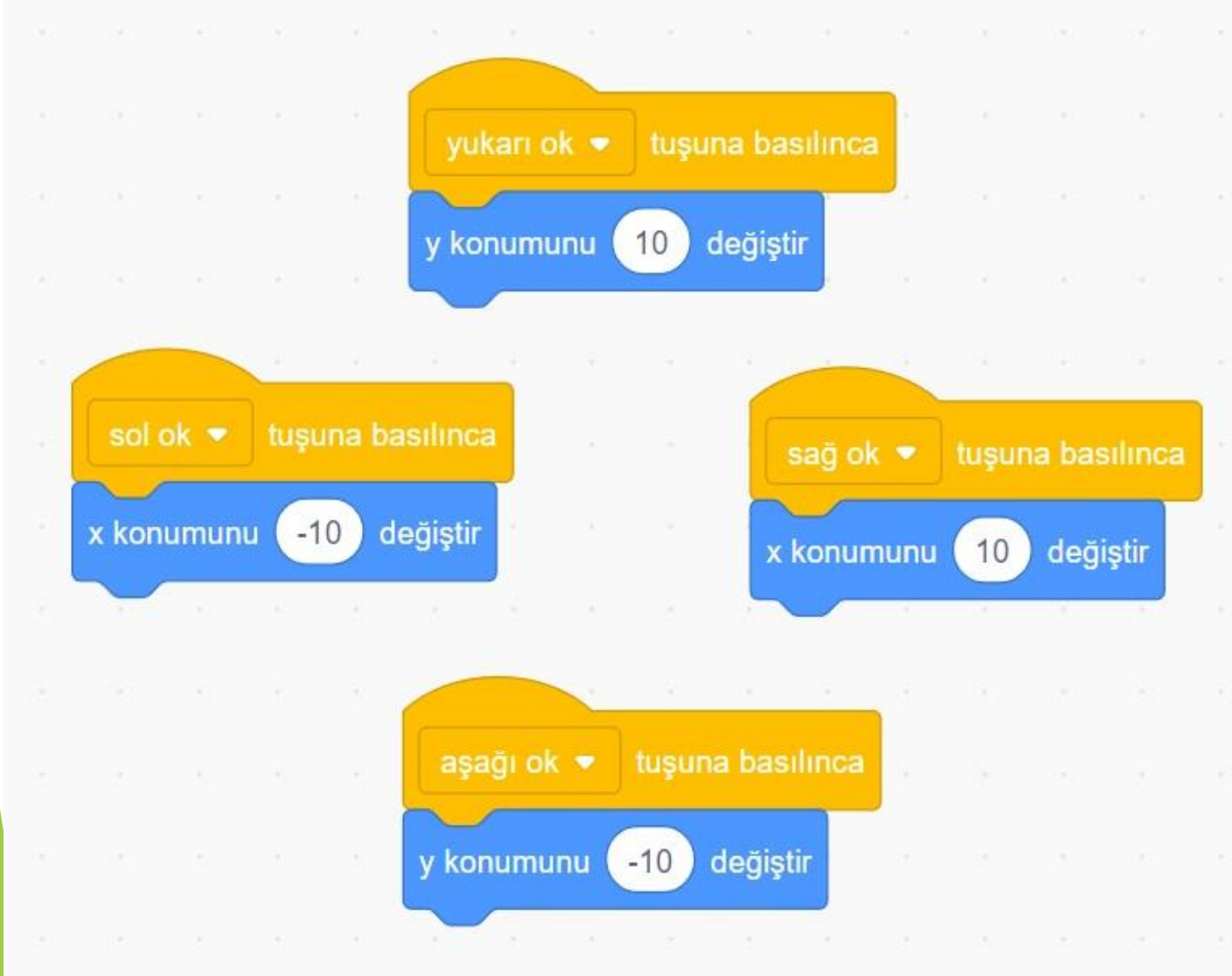
Tam orta noktasında olmasını istiyorsak (0,0) noktasına getirmeliyiz.

X değeri artarsa karakterimiz sağa, azalırsa sola hareket eder. Y değeri artarsa karakterimiz yukarı, azalırsa aşağı yönde hareket eder.

Çapraz yönlerde hareket ettirmek istersek x ve y değerlerini aynı anda artırmalı veya azaltmalıyız.

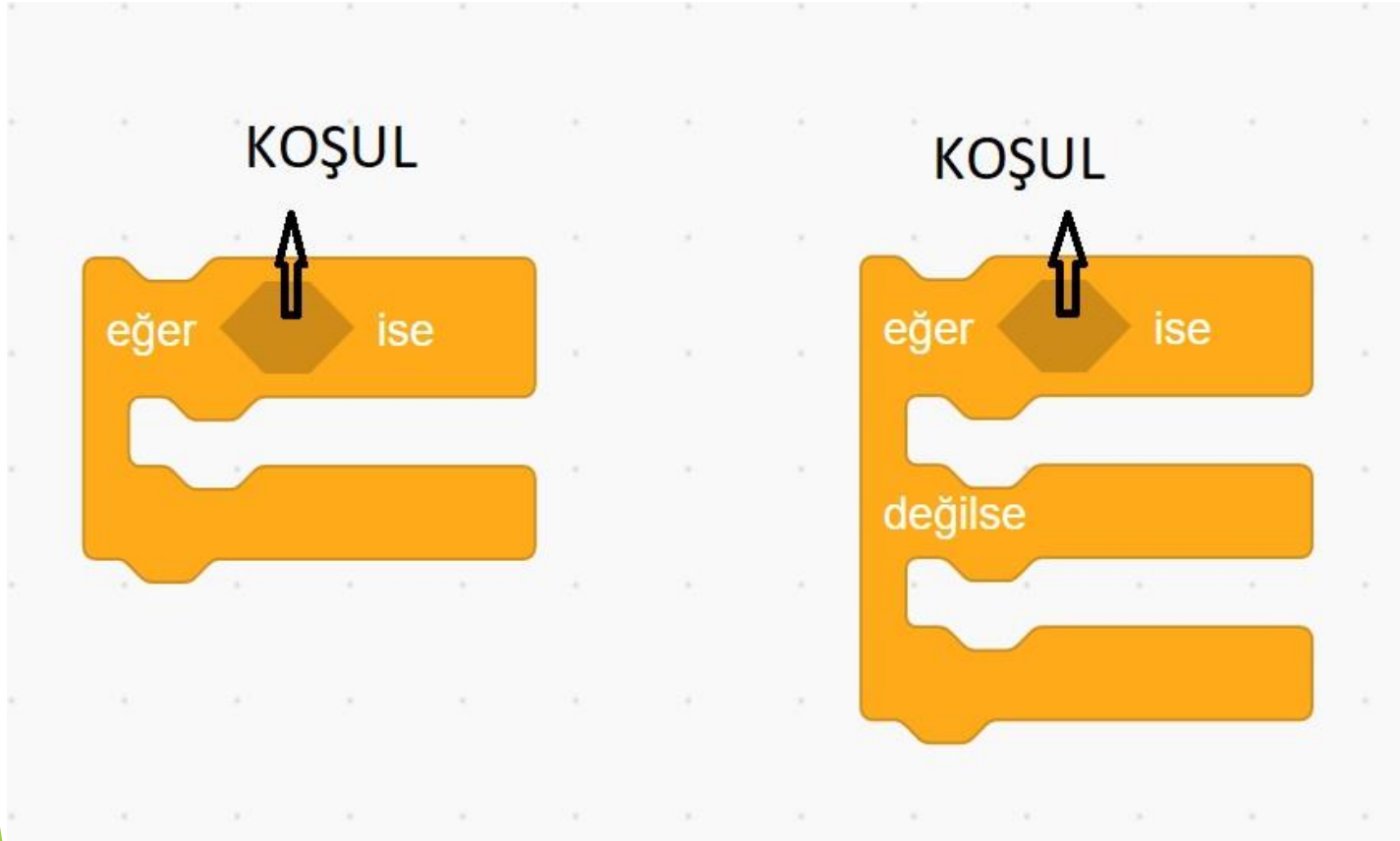


e) Karakteri Yön Tuşları İle Hareket Ettirme

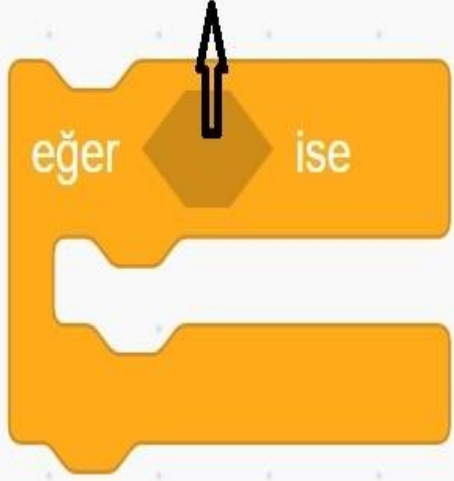


f) Karar Mantık Yapısı

- Bilgisayara iki yada daha fazla seçenek arasından seçim yaptırmak için kurulan mantık yapılarıdır.



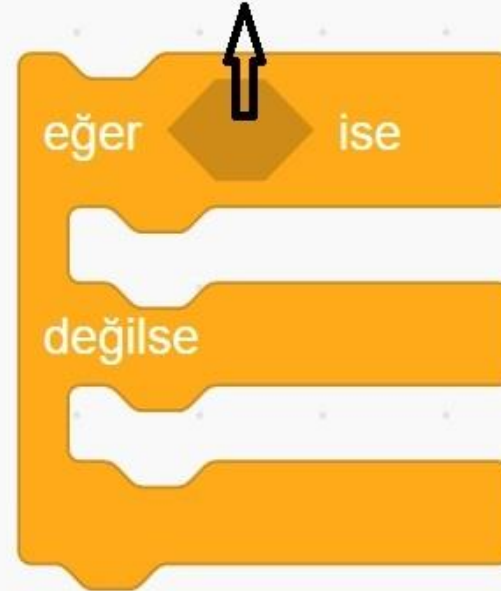
KOŞUL



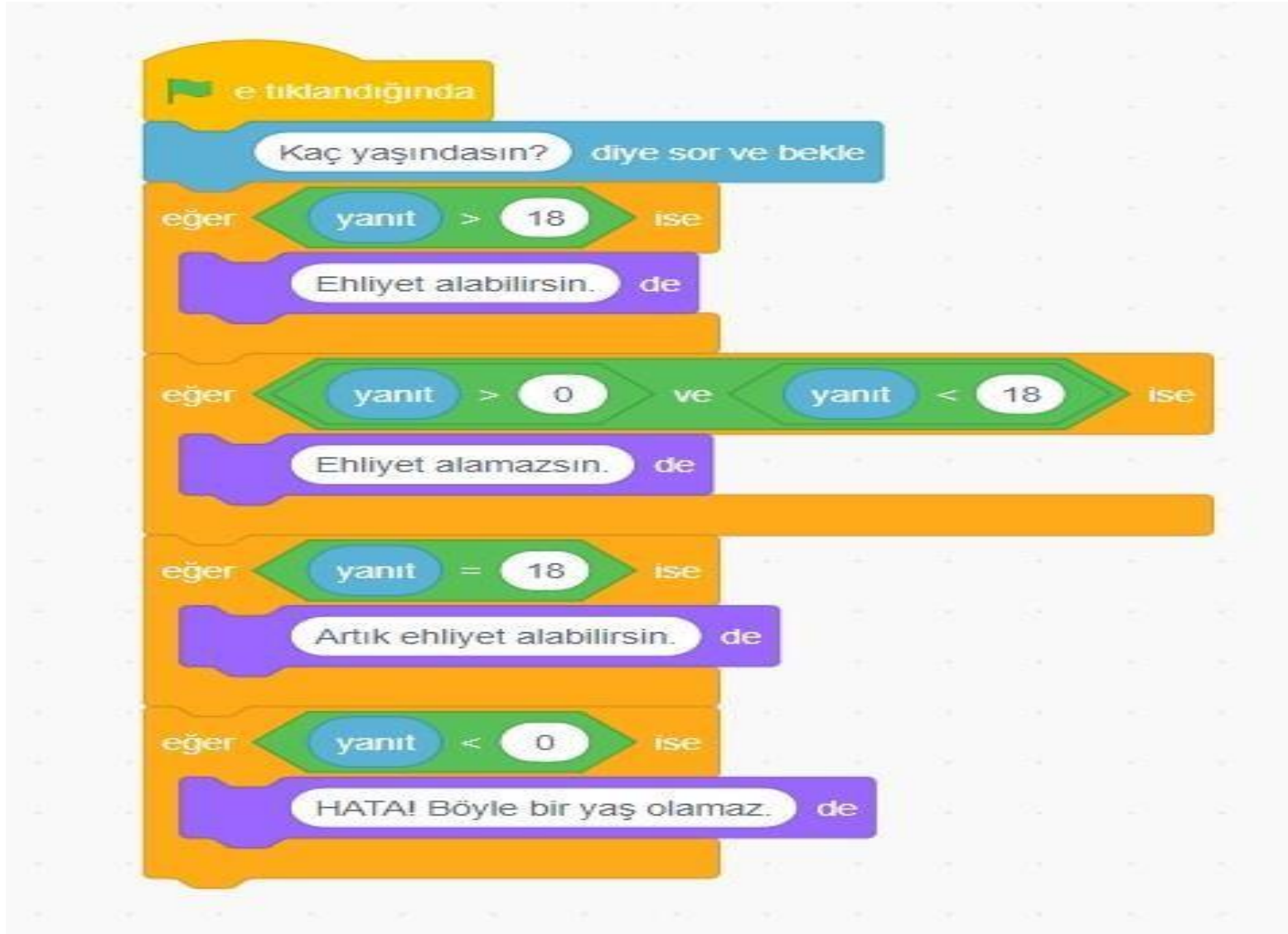
Yazılan koşul **doğru** ise içerisindeki blokları çalıştırır, **yanlış** ise blokları çalıştırmadan alttaki komutlardan devam eder.

Koşul gerçekleşir ise ilk boşlukta yer alan bloklar çalışır, gerçekleşmez ise ikinci boşlukta yer alan bloklar çalışır.

KOŞUL



6) Ehliyet Alabilir Misin?



g) Değişkenler

- ▶ Program, oyun sürecinde girilen sayı, yazı, doğru-yanlış, tarih vb. değerleri tutan programlama birimleridir.
- ▶ Değişkenleri biz tanımlarız ve isim veririz.
- ▶ Aynı isimde birden fazla değişken olamaz.

Değişkenler

Bir Değişken Oluştur

değişkenim

değişkenim 'i 0 yap

değişkenim 'i 1 kadar değiştir

değişkenim 'i değişkenini göster

değişkenim 'i değişkenini gizle

h) Sayaç



Yeşil bayrak tıklandığında program başlayarak her bir saniyede süre değişkenini 1 artırır. Bu sayede sayaç/kronometre programlanmış olur.

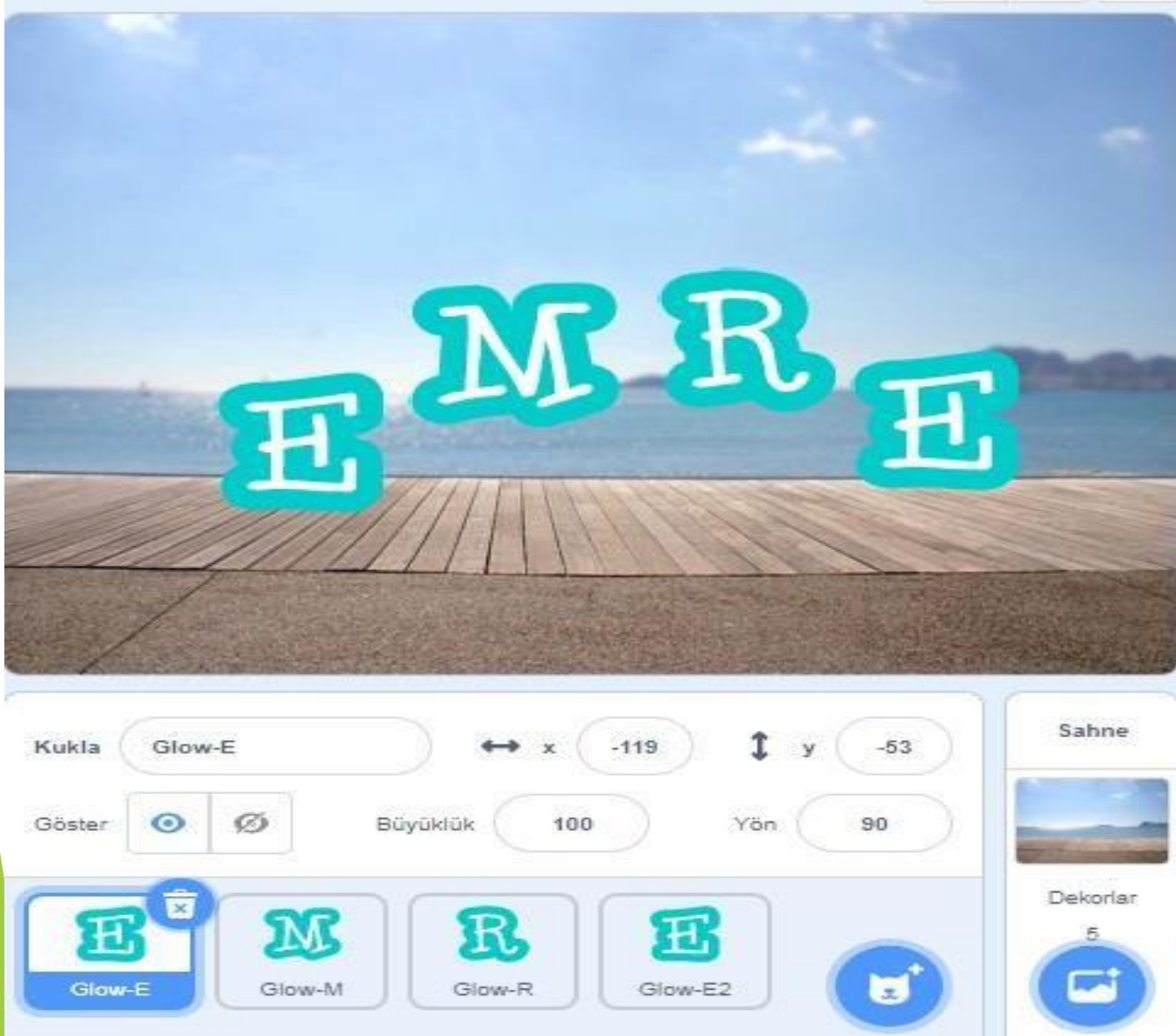
Proje 1) Kuklamızı Hareket Ettirme



Proje 2)Kuklaları Konuşturma

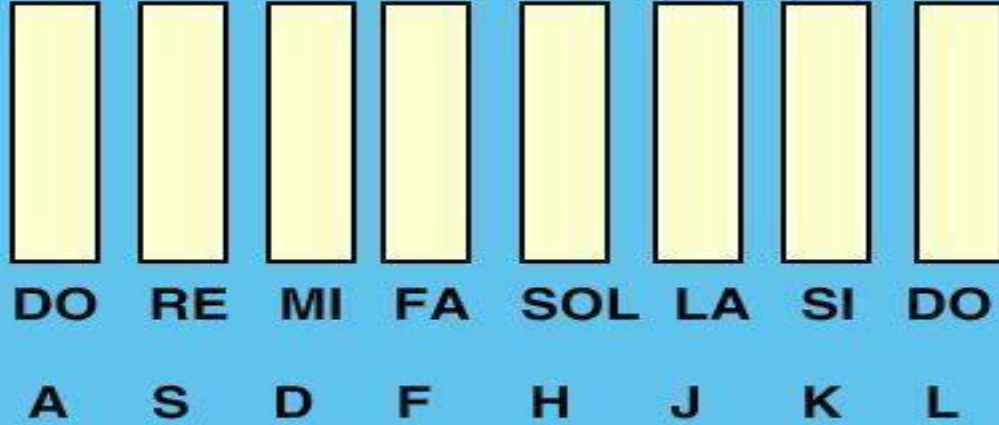


Proje 3) İsmimi Canlandır



Proje 4) Piyano Yapalım

PIYANO CALIYORUM



Kuklalar

Yeni kukla:



Kukla2



Kukla3



Kukla4



Kukla5



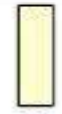
Kukla6



Kukla7



Kukla8



Kukla9

Yeni dekor:



bu kukla tıklanınca

x: -125 y: 30 noktasına git

60 notasını 0.5 vuruş çal

a tuşu basınca

60 notasını 0.5 vuruş çal

s tuşu basınca

62 notasını 0.5 vuruş çal

d tuşu basınca

64 notasını 0.5 vuruş çal

f tuşu basınca

65 notasını 0.5 vuruş çal

h tuşu basınca

67 notasını 0.5 vuruş çal

j tuşu basınca

69 notasını 0.5 vuruş çal

k tuşu basınca

71 notasını 0.5 vuruş çal

l tuşu basınca

72 notasını 0.5 vuruş çal

Proje 5) Süre / Sayaç Yapımı



Aşağıdaki kelimeleri tanımlardaki uygun yerlere yerleştiriniz.

Problem

Akış Şeması

Yazılım

Algoritma

-: Çeşitli görevleri gerçekleştirmek amacıyla hazırlanmış programlara denir.
-: Çözülmesi gereken sorun yada aşılması gereken engeldir.
-: Bir problemin çözümünde izlenecek yol anlamına gelir ve problemin adımlar halinde yazılması ile oluşur.
-: Bilgisayar programlarının işlem basamaklarınınin geometrik şekillerle gösteren şemadır.

Aşağıdaki problemin algoritmasını liste halinde yanına yazınız.

KURT, KUZU VE OT

Ahmet Amca'nın çiftliği köyün biraz dışında Kızıldere'nin hemen öbür yanındaymış. Ahmet Amca bir gün kuzusunu, ormandan bahçesine inen kurdu ve kuzusu için ayırdığı bir miktar otu da alıp karşı kıyıya geçmek istemiş. Ancak karşıya geçebileceği tek araç ufacık bir kayıkmış ve hepsinin beraber karşıya geçmesi imkansızmış. Kayığa her defasında birini alabiliyormuş; ya kuzuyu ya kurdu ya da otu yanına alabilecekmış. Ancak bir sorunu daha varmış, kurtla kuzuyu yalnız bırakırsa kurt kuzuyu yermiş, kuzuyla otu yalnız bıraksa bu sefer kuzu da otları yermiş. Peki sizce nasıl olacak dva Ahmet Amca üçünü birden karşıya geçirecek?

1)

2)

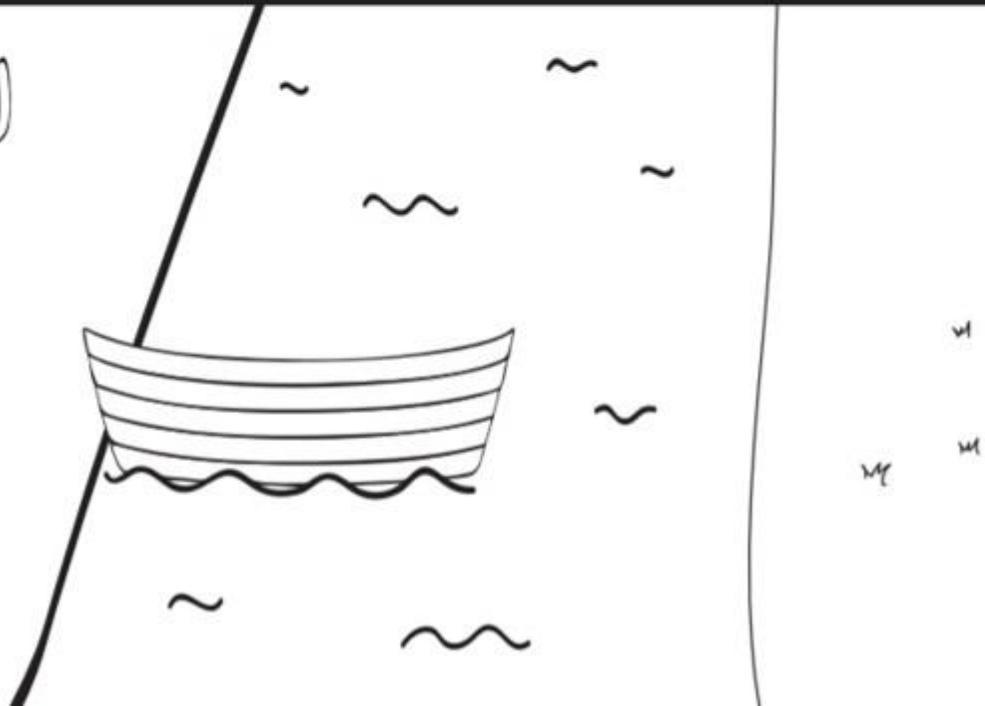
3)

4)

5)

6)

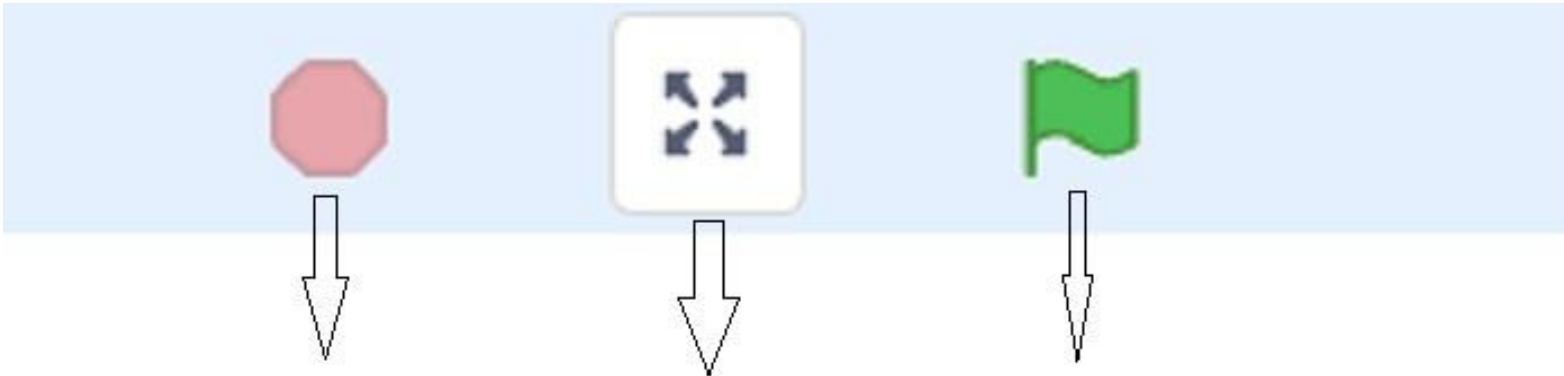
7)



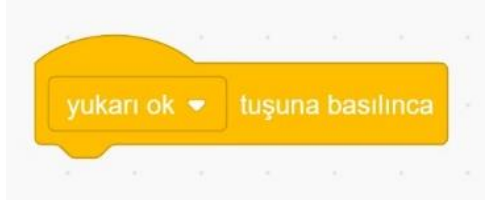
Aşağıda yer alan 2 döngü bloğu arasındaki farkı yazınız.



Aşağıda yer alan butonların görevlerini altlarına yazınız.



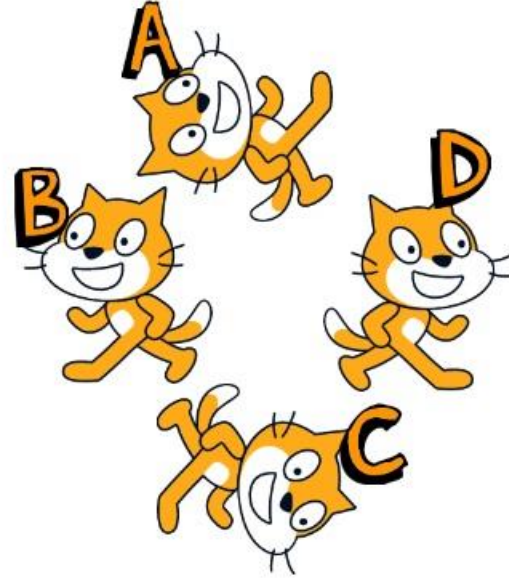
Aşağıda yer alan kod bloklarının görevlerini yanlarına yazınız.



Kedi karakterimizi koşuyormuş gibi göstermek için aşağıda yer alan kod bloklarında soru işaretli olan bölgeye hangi blok gelmelidir?



Aşağıda yer alan karakterlerin sahnede hangi yöne baktıklarını göstermek için kullandığımız açılar yazınız.



x konumunu 10 deęiřtir

y konumunu 10 deęiřtir

y konumunu -10 deęiřtir

x konumunu -10 deęiřtir

Yukarıda yer alan blokları kullanarak, klavyede bulunan yön tuřları ile karakterimizi hareket ettirmek için gereken blokları uygun yerlerine yerleřtiriniz.

yukarı ok ▼ tuřuna basılınca

sol ok ▼ tuřuna basılınca

saę ok ▼ tuřuna basılınca

ařaęı ok ▼ tuřuna basılınca

Aşağıda yer alan bloklar ayrı ayrı çalıştırıldığında karakterimiz hangi yöne hareket eder.

x konumunu 10 değiştir

y konumunu -10 değiştir

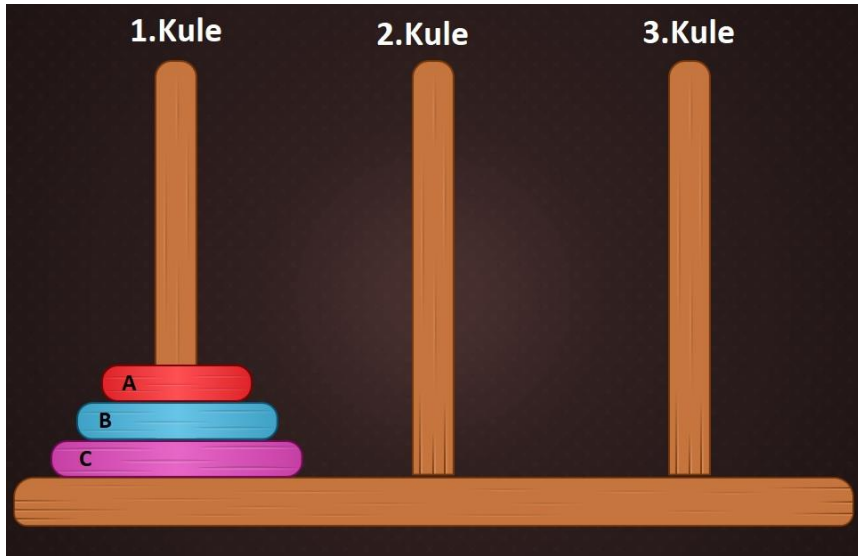
y konumunu 10 değiştir

x konumunu -10 değiştir

Aşağıda yer alan döngü bloklarının görevlerinin yazınız.



Aşağıda yer alan hanoi kuleleri probleminin çözüm algoritmasını liste halinde yazınız.



- 1)
- 2)
- 3)
- 4)
- 5)
- 6)
- 7)