

PYTHON İLE TANILAYICI DALLANMIŞ AĞAÇ MODELİ

Arş. Gör. Nisa Yenikalaycı
Ondokuz Mayıs Üniversitesi
nisa.yenikalayci@omu.edu.tr

Özet

Çalışmanın amacı, alternatif ölçme ve değerlendirme araçlarından biri olan tanılayıcı dallanmış ağaç modelini bilgisayar üzerinde Python programlama dilini kullanarak modelleyebilmektir. Verilerin oluşturulması amacıyla, ortaöğretim fizik 11. sınıf ders kitabında "kuvvet ve hareket" ünitesinde yer alan bir tanılayıcı dallanmış ağaç modeli örnek olarak seçilmiştir. Kodların oluşturulması amacıyla, Python 3.5.1 sürümü kullanılmıştır. Oluşturulan kodların doğruluğuna ilişkin, programlama dilleri konusunda yeterli bilgisi olan bir uzmanın görüşü alınmıştır. Programı çalıştıran bir kullanıcı, ilk çıkan ifadeye verdiği yanıtla göre doğru veya yanlış olarak ilerleyerek bir çıkış noktasından çıkacaktır. İfadelere yanıt olarak büyük (D-Y) veya küçük (d-y) harfleri girilebilir. Kullanıcı bunların dışında bir ifade ya da karakter girerse "D veya Y harfleri ile giriş yapınız_!" uyarısı ekrana gelecektir. Eğer kullanıcı programa yanıt olarak baştan hatalı bir ifade girerse programın çalışması duracaktır. Çıkış noktasından aldığı puan o kişinin konuyu ne derecede kavradığını göstermesi bakımından önemlidir. Bu çalışma süre tutularak yapılabilir, öğrencilerin uygulamada doğru cevaba ulaşma süresine bakılabilir. Öğrencilerden kodlara ait veri akışından yola çıkarak tanılayıcı dallanmış ağaca ait yapıyı kendilerinin oluşturulmaları istenebilir.

Anahtar Sözcükler: Fen eğitimi, alternatif ölçme ve değerlendirme, tanılayıcı dallanmış ağaç, Python.

DIAGNOSTIC TREE MODEL WITH PYTHON

Abstract

The goal of this study is to model diagnostic tree model which is one of alternative measurement and assessment means with the use of Python programming language on computer. In the purpose of generating data, a diagnostic tree model which was in the unit of "power and move" in 11rd grade's course book of secondary education physics was chosen as a sample. In the purpose of generating codes, Python 3.5.1 version was used. Regarding the accuracy of codes which are generated, a view of a specialist that he has got sufficient knowledge on programming languages was taken. A user who work the program will come of the starting point in processing correctly or incorrectly by the answer that he gives on the first expression. Block letter (T/F) or lower letter (t/f) can be entered for expressions as response. If the user enters an expression or character except from them, the warning "please, press on the button T or F_!" will come on the screen. If the user enters an incorrect expression at the beginning as a response to the program, the running of program will stop. The point that the person gets from the starting point is important in terms of indicating on what an extent he/she perceives the subject. This study can be done by timing, the time that students have got to get correct answer in the practice can be considered. Students can be asked to form the structure belonging to the diagnostic tree on their own on the basis of data flow belonging to codes.

Keywords: Science, alternative measurement and assessment, diagnostic tree, Python.

GİRİŞ

Bu çalışmanın amacı, alternatif ölçme ve değerlendirme araçlarından biri olan tanılayıcı dallanmış ağaç modelini bilgisayar üzerinde Python programlama dilini kullanarak modelleyebilmektir. Çalışmadaki bağımsız değişken, Python komut satırında verilen yanıtlar (D-Y veya d-y); bağımlı değişken, tanılayıcı dallanmış ağaçtan çıkış

noktası olarak ifade edilebilir. Sabit değişkenler ise, örnek olarak seçilen tanılayıcı dallanmış ağaç modeli ve bu modelin ilk ifade cümlesi olarak belirtilebilir.

Çakallı (2008) ilköğretim okulu yöneticilerinin eğitimde bilgisayar kullanımına yönelik karşılaştıkları sorunlar ve bilgisayar tutumlarını belirlemek amacıyla yaptığı çalışmada, mevcudu 50'nin altında olan hiçbir okulda bilgisayar laboratuvarı olmadığı sonucuna ulaşmıştır. Laboratuvarsız bir eğitim ortamında bilgisayar kullanımına karşı olumlu tutumlar geliştirmenin zor olacağı ifade edilebilir. Öğrencilerin bu konuda olumlu tutum geliştirmelerinde öncülük edecek olan öğretmen ve yöneticilerin iyi düzeyde bilgisayar okuryazarı olmaları arzu edilen bir durumdur. Benzer şekilde Kocasaray (2003) bilgisayarların öğretim alanında kullanımına ilişkin öğretmen yeterliliklerinin belirlenmesi amacıyla yaptığı çalışmada, öğretmen yeterliliğine en çok etki eden faktörler arasında; cinsiyet, bilgisayar kursu süresi, öğretmenin yaşı, branşı ve bilgisayarı derslerinde kullanıp kullanmaması olduğu sonucuna ulaşmıştır.

Usluel ve Seferoğlu (2003) çalışmada, eğitim fakültesinde görevli öğretim elemanlarının bilgisayar kullanma öz-yeterlik algıları ile bilgisayar kullanımı arasında anlamlı bir ilişki olduğu sonucuna ulaşmıştır. Öz-yeterlik algısının yükselmesi için; alt yapı, donanım ve teknik desteğin sağlanmasına ihtiyaç duyulduğu söylenilebilir.

Öğretmenler bilgisayarları ölçme değerlendirme aracı olarak kullanabilmelidirler. Ölçme değerlendirme etkinlikleri, içerik ve hedeflere uygun olmalıdır. Alternatif ölçme ve değerlendirme teknikleri, geleneksel değerlendirmelerin dışında kalan tüm değerlendirme türlerini kapsar (Milli Eğitim Bakanlığı Talim ve Terbiye Kurulu [MEB TTK], 2005:24). Bu çalışmada alternatif ölçme ve değerlendirme tekniklerinden tanılayıcı dallanmış ağaç modeli kullanılmıştır. Tanılayıcı dallanmış ağaç, belli bir konuda öğrencinin neleri öğrendiğini ve neleri öğrenemediğini belirlemek için kullanılan değerlendirme araçlarından biridir. Bu teknikte temelden ayrıntıya giden bir sıraya göre doğru ve yanlış ifadeler seçilerek öğrenciden doğru seçimi yapması istenir. Böylece 8 veya 16 seçimli bir ifadeler listesi ile sonlanan bir dallanmış ağaç oluşturulur (MEB TTK, 2005:40). Bu çalışmada 8 seçimli model kullanılmıştır.

Kocaarslan (2012) ilköğretim programlarını değerlendirme çalışmalarında, öğretmenlerin tanılayıcı dallanmış ağaç tekniği ilgili yeterli bilgilerinin olmadığını ve bu tekniğin alanda istenilen düzeyde kullanılmadığı sonucuna ulaşmıştır. Karahan (2007) kavram haritası, grid ve dallanmış ağaç gibi metodların alternatif ölçme ve değerlendirme yöntemleri olarak ortaöğretim 9. sınıf biyoloji dersi konularında uygulanabilirliği ve öğrenci başarısına katkısını belirlemek amacıyla yaptığı çalışmada, bu tekniklerin biyoloji konularına uygulanabildiği ve öğrenci başarısına katkıda bulunduğu sonucuna ulaşmıştır. Benzer şekilde Öztürk (2011) çalışmada, kavram haritası, grid ve dallanmış ağaç gibi alternatif ölçme ve değerlendirme tekniklerinin ilköğretim 8. sınıf fen ve teknoloji konularında uygulanabilirliği ve öğrenci tutumuna etkisini incelemek amacıyla yaptığı çalışmada, öğrencilerin fen ve teknoloji dersine karşı tutumlarında olumlu yönde bir değişim olduğu sonucuna ulaşmıştır. Yine benzer şekilde, Polat (2011) çalışmada, vee diyagramı, tanılayıcı dallanmış ağaç ve kavram haritalarının matematik dersine yönelik tutum ile başarıya etkilerini incelemek ve bu araçlara ilişkin öğretmen görüşlerini belirlemek amacıyla yaptığı çalışmada, öğretmenlerin programlarda önerilen yeni ölçme ve değerlendirme araçları ile ilgili olumlu görüşlere sahip olduğu sonucuna ulaşmıştır.

Bilgisayarı eğitim amaçlı kullanmada önemli bir parça olan ölçme ve değerlendirme kısmının bilgisayar uygulamalarına entegre edilebilmesi oldukça önemlidir. Alanyazın tarandığında, Python programlama dili ile tanılayıcı dallanmış ağacı birleştiren bir çalışmaya rastlanmamıştır. Bilgisayarlar, programlama dilleri aracılığıyla; bilim insanları gibi karmaşık sistemlerin davranışlarını gözlemlerler, hipotezler oluştururlar ve varsayımları sınarlar (Elkner, Dawney & Meyers, 2008:7). Bu nedenle öğretmenlerin temel düzeyde programlama öğrenmeleri gerekmektedir.

Python; web tabanlı uygulamalar, ağ, görsel arayüzler, işletim sistemi ile ilgili uygulamalar, oyunlar, veri işleme, iş hayatı için uygulamalar vb. birçok alanda kullanılmaktadır (Karaçay, 2014:8). Python'un en temel nitelikleri şu şekilde ifade edilebilir: Python kendi web sayfasından ücretsiz olarak indirilebilir, kaynak programın yazılması ve okunması çok kolaydır, kolay öğrenilir ve kolay kullanılır (Karaçay, 2014:6-7).

Bu çalışmada "1) komut satırından girdi: input, 2) koşullu işletme: if deyimi, 3) if-else deyimi" olmak üzere 3 temel komut kullanılmıştır (Başer, 2015; Özgül, 2010; Briggs, 2013). Örnekte yer alan "Bir sistem, kendine etki

eden kuvvetler $R=0$ ve $\Sigma\tau=0$ ise dengededir." ifadesi doğru bir ifadedir ve programı çalıştıran bir kullanıcının "D" veya "d" olarak yanıt verip tanılayıcı dallanmış ağaç sürecine devam etmesi beklenmektedir. Kodlara ait veri akışından yola çıkarak tanılayıcı dallanmış ağaca ait yapıyı öğrencilerden kendilerinin oluşturmaları istenebilir. Örneğin bu kodları çalıştıran bir öğrenci, programdan aldığı yanıtlara göre şemanın nasıl olması gerektiğini hayal eder ve tanılayıcı dallanmış ağaca ait iskeleti oluşturabilir. Böylelikle öğrencilerin düşünme becerilerinin gelişimine katkı sağlayacağı düşünülmektedir.

YÖNTEM

Bu çalışma ile bilgisayarda programlama dili ile alternatif ölçme ve değerlendirme yaklaşımı arasında bağ kurulmak istenmiştir. Bu amaçla, ortaöğretim fizik 11. sınıf ders kitabında "kuvvet ve hareket" ünitesinde yer alan bir tanılayıcı dallanmış ağaç modeli örnek olarak seçilmiştir (Kurnaz, Değermenci, Kalyoncu, Pektaş, Bayraktar, Aydın & Moradaoğlu, 2012:147). Burada seçilen örneğin bir ayrıcalığı bulunmamaktadır. Tanılayıcı dallanmış ağacın çalışma prensibini doğru kodlarla oluşturabilmek hedeflenmiştir.

Örnekleme

Tipik durum örnekleme, araştırma problemi ile ilgili olarak evrende yer alan çok sayıdaki durumdan tipik olan bir durumun belirlenerek bu örnek üzerinden bilgi toplanmasını gerektirir (Büyüköztürk, Kılıç Çakmak, Akgün, Karadeniz, Demirel, 2014:91).

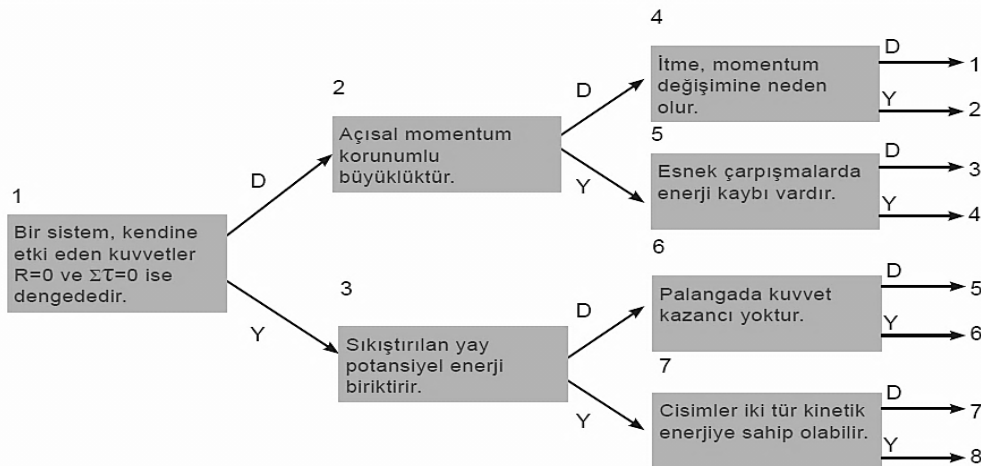
Verilerin Toplanması

Olgusal nitelikte veriler kullanılmıştır. Bunlar; kişisel yargılardan bağımsız olarak var olan, herkesin üzerinde anlayabildiği türden gözlenebilir ölçütleri olan gerçeklerdir. Cinsiyet, yaş, boy, ağırlık gibi (Karasar, 2013:132). Verilerin oluşturulması amacıyla Windows işletim sistemi için uyumlu olan Python 3.5.1 sürümü kullanılmıştır. Python yazılımı 32 bit işletim sisteminde çalıştırılmıştır. Python programı kendi sitesinden kolaylıkla indirilebilmektedir. Bu programlama dilinin seçilme nedeni, öğrenilmesinin ve kullanılmasının kolay olmasıdır. Oluşturulan kodların doğruluğuna ilişkin programlama dilleri ile ilgili bilgisi olan bir uzmanın görüşü alınmıştır. Doğru kodlarla oluşturulan programı çalıştıran bir kullanıcı, ilk çıkan ifadeye verdiği yanıtlara göre doğru veya yanlış olarak ilerleyecek ve bir çıkış noktasından çıkacaktır. Öğrencilerin yanıtlarına göre bu çıkışlara puan verilebilir. Örneğin; 1 nolu çıkış 100 puan gibi.

Verilerin Analizi

Çıkış noktasından aldığı puan, öğrencinin konuyu ne derecede kavradığını göstermesi bakımından önemlidir. Eğer tanılayıcı dallanmış ağacı kendimiz hazırlayacak olsak ifadelerin yönlendirici olmamasına dikkat etmeliyiz.

Burada öğretmen yeterliliklerine değinilerek, öğretmenler tarafından temel düzeyde mantıksal akış diyagramının oluşturulabilmesi konusuna vurgu yapılmıştır. Seçilen tanılayıcı dallanmış ağaç modeli Şekil 1'de sunulmuştur.



Şekil 1:Seçilen Tanılayıcı Dallanmış Ağaç Modeli

Kaynak : Kurnaz, Değermenci, Kalyoncu, Pektaş, Bayraktar, Aydın & Moradaoğlu, 2012:147.

Şekil 1’de görüldüğü gibi, ağaç dallara ayrılarak devam etmektedir ve bu yollardan bazıları doğru bazıları yanlıştır. Tanılayıcı dallanmış ağaç tekniğinden, öğrenme-öğretme sürecinde, tanılayıcı bir işlevle öğrencilerdeki kavram yanlışlarını ortaya çıkarmak için ya da öğrenme sürecinin sonucunda öğrencileri değerlendirmek için yararlanılabilir (Kocaarslan, 2012:274). Tanılayıcı dallanmış ağaca ait verilen yanıtlar ve çıkışlar Tablo 1’de sunulmuştur.

Tablo 1: Tanılayıcı Dallanmış Ağaca Ait Verilen Olası Yanıtlar ve Çıkışlar

Verilen Yanıtlar	Çıkışlar
1 (D), 2 (D), 4 (D)	1
1 (D), 2 (D), 4 (Y)	2
1 (D), 2 (Y), 5 (D)	3
1 (D), 2 (Y), 5 (Y)	4
1 (Y), 3 (D), 6 (D)	5
1 (Y), 3 (D), 6 (Y)	6
1 (Y), 3 (Y), 7 (D)	7
1 (Y), 3 (Y), 7 (Y)	8

Tablo 1’de görüldüğü gibi, verilen yanıtlara göre çıkış noktaları "1, 2, 3, 4, 5, 6, 7 ve 8" olarak belirtilmiştir. Python’da oluşturulan tanılayıcı dallanmış ağaca ait kodlar Ek’te sunulmuştur. Ek’te görüldüğü gibi, bütün olası durumlara ait kodlar oluşturularak çıkış noktaları belirlenmiştir. Program dönüt (geribildirim) sağlar. Bir ifadeye doğru (D-d) veya yanlış (Y-y) olarak girildiğinde bir sonraki basamağa geçip sınanacak yeni bir ifade yazmaktadır. Fakat kullanıcı doğru veya yanlışın dışında bir ifade ya da karakter girerse "D veya Y harfleri ile giriş yapınız_!" uyarısı ekrana gelecektir. Yanıt olarak büyük (D-Y) veya küçük (d-y) girilebilir. Eğer kullanıcı, programa yanıt olarak baştan hatalı bir ifade girerse programın çalışması durmaktadır.

SONUÇ VE ÖNERİLER

Python, öğrencilerin bilgisayarlı çalışmalarında daha iyi bir başlangıç yapabilmelerine yardımcı olacaktır (Malkoç, 2012:201). Bilgisayar derslerinin içerisine öğretmen adayının bilgisayardan eğitim amaçlı nasıl yararlanabileceğine ilişkin olarak Python programlama dilinin genel çerçevesiyle öğretilmesi konusunda müfredata eklemeler yapılabilir. Üniversitelerin eğitim fakültelerinde; Python’a ek olarak Perl, PHP, Java Script, HTML, ASP gibi programlama dilleri de seçmeli ders olarak öğretilmelidir. Bu tür programlama uygulamalarının geleceğin öğretmenlerine kendi ders materyallerini oluşturmada katkı sağlayacağı düşünülmektedir.

Tanılayıcı dallanmış ağaç modelinin kodlarının oluşturulduğu bu çalışma, süre tutularak da yapılabilir. Bir bilgisayar laboratuvarına alınan öğrencilerin doğru cevaba ulaşma süreleri kaydedilebilir.

Başka alternatif ölçme ve değerlendirme yaklaşımlarında bu programlama dili kullanılabilir. Örneğin, bu konuda uzman görüşleri alınarak hazırlanan rubrik (dereceli puanlama anahtarı) kullanılarak yapılan değerlendirmeler için de kodlar oluşturulabilir.

Teşekkür: Katkılarından dolayı arkadaşım Ülkü Kocaoğlu’na teşekkür ederim.

Not: Bu çalışma 27- 29 Ekim 2016 tarihlerinde Antalya’da 7 Ülkenin katılımıyla düzenlenen World Conference on Educational and Instructional Studies- WCEIS’de bildiri olarak sunulmuştur.

KAYNAKÇA

Başer, M. (2015). *Python*. (3. b.). İstanbul: Dikeyksen Yayıncılık.

Briggs, J. R. (2013). *Python for kids - A playful introduction to programming*. San Francisco: No Starch Press.

Büyüköztürk, Ş, Kılıç Çakmak, E., Akgün, Ö. E., Karadeniz, Ş. & Demirel, F. (2014). *Bilimsel Araştırma Yöntemleri* (18. b.). Ankara: Pegem Akademi Yayınları.

Çakallı, A. (2008). *İlköğretim okulu yöneticilerinin bilgisayar tutumları ve eğitimde bilgisayar kullanımına yönelik karşılaştıkları sorunlar (Samsun ili örneği)*. Yüksek Lisans Tezi, Yeditepe Üniversitesi, Eğitim Yönetimi ve Denetimi Anabilim Dalı, İstanbul.

Elkner, J., Dawney, A. B. & Meyers, C. (2008). *Bilgisayara bilimcisi gibi düşünmek (Python ile öğrenme)*. (2. b.). (T. E. Kalaycı, Çev.)

Karaçay, T. (2014). *Python 3 veri yapıları*. (2. b.). Ankara: Seçkin Yayıncılık.

Karahan, U. (2007). *Alternatif ölçme ve değerlendirme metodlarından grid, tanılayıcı dallanmış ağaç ve kavram haritalarının biyoloji öğretiminde uygulanması*. Yüksek Lisans Tezi, Gazi Üniversitesi, Eğitim Bilimleri Anabilim Dalı, Ankara.

Karasar, N. (2013). *Bilimsel araştırma yöntemi*. (25. b.). Ankara: Nobel Akademik Yayıncılık.

Kocaarslan, M. (2012). Tanılayıcı dallanmış ağaç tekniği ve ilköğretim 5. sınıf fen ve teknoloji dersi maddenin değişimi ve tanınması adlı ünite kullanımı. *Mustafa Kemal Üniversitesi Sosyal Bilimler Enstitüsü Dergisi*, 9(18), 269-279.

Kocasaraç, H. (2003). Bilgisayarların öğretim alanında kullanımına ilişkin öğretmen yeterlikleri. *The Turkish Online Journal of Educational Technology - TOJET*, 2(3).

Kurnaz, Değermenci, Kalyoncu, Pektaş, Bayraktar, Aydın & Moradaoğlu. (2012). *Ortaöğretim fizik 11 ders kitabı*. (3. b.). Ankara: Başak Matbaacılık.

Malkoç, B. (2012). Temel bilimler ve mühendislik eğitiminde programlama dili olarak Python. *XIV. Akademik Bilişim Konferansı*. Uşak Üniversitesi.

Milli Eğitim Bakanlığı Talim ve Terbiye Kurulu Başkanlığı [MEB TTK]. (2005). *İlköğretim fen ve teknoloji dersi öğretim programı ve klavuzu 6-7-8. sınıflar (taslak basım)*. Ankara.

Özgül, F. (2010). *Python*. (1. b.). İstanbul: Kodlab Yayınları.

Öztürk, P. T. (2011). *İlköğretim 8. sınıf "Canlılar ve enerji ilişkileri" ünitesinin kavram haritaları, yapılandırılmış grid ve tanılayıcı dallanmış ağaç teknikleri ile işlenmesinin öğrencilerin fen ve teknoloji dersine karşı tutumları üzerine etkisi*. Yüksek Lisans Tezi, Selçuk Üniversitesi, İlköğretim Anabilim Dalı, Konya.

Polat, B. (2011). *Vee diyagramı, tanılayıcı dallanmış ağaç ve kavram haritalarının matematik dersine yönelik tutum ile başarıya etkileri ve bu araçlara yönelik öğretmen görüşleri*. Yüksek Lisans Tezi, Hacettepe Üniversitesi, Eğitim Bilimleri Anabilim Dalı, Ankara.

Usluel, Y. & Seferoğlu, S. S. (2003). Eğitim fakültelerindeki öğretim elemanlarının bilgisayar kullanımı ve öz-yeterlik algıları. *Bilişim Teknolojileri Işığında Eğitim Konferansı - BTIE*. Ankara: ODTÜ Kültür ve Kongre Merkezi.

EK: Python'da Oluşturulan Tanılayıcı Dallonmuş Ağaca Ait Kodlar

```
print ("İfadelerin Doğru ya da Yanlış olduğunu belirtmek için D veya Y tuşuna basınız...")
print()
Soruyanlis=0
while Soruyanlis==0:
    Soru=input ("Bir sistem, kendine etki eden kuvvetler  $R=0$  ve  $\Sigma\tau=0$  ise dengededir.")
    print()
    if Soru=="d" or Soru=="D":
        D_1yanlis=0
        while D_1yanlis==0:
            D_1=input ("Açıl momentum korunumu büyüktür.")
            print()

            if D_1=="d" or D_1=="D":
                D_1_1yanlis=0
                while D_1_1yanlis==0:
                    D_1_1=input ("İtme, momentum değişimine neden olur.")
                    print()
                    if D_1_1=="d" or D_1_1=="D":
                        print("1 Nolu Çıkış...")
                        D_1_1yanlis=1
                        break
                    elif D_1_1=="y" or D_1_1=="Y":
                        print("2 Nolu Çıkış...")
                        D_1_1yanlis=1
                        break
                    else:
                        print("D veya Y harfleri ile giriş yapınız_!")
                        print()
                        D_1_1yanlis=0
                D_1yanlis=1
            break

        elif D_1=="y" or D_1=="Y":
            D_1_2yanlis=0
            while D_1_2yanlis==0:
                D_1_2=input("Esnek çarpışmalarda enerji kaybı vardır.")
                print()
                if D_1_2=="d" or D_1_2=="D":
                    print("3 Nolu Çıkış...")
                    D_1_2yanlis=1
                    break
                elif D_1_2=="y" or D_1_2=="Y":
                    print("4 Nolu Çıkış...")
                    D_1_2yanlis=1
                    break
                else:
                    print("D veya Y harfleri ile giriş yapınız_! ")
                    print()
                    D_1_2yanlis=0
            D_1yanlis=1
        break
```

```
else:
    print("D veya Y harfleri ile giriş yapınız_!")
    print()
    D_1yanlis=0
Soruyanlis=1
break

elif Soru=="y" or Soru=="Y":
    Y_1yanlis=0
    while Y_1yanlis==0:
        Y_1=input("Sıkıştırılan yay potansiyel enerji biriktirir.")
        print()

        if Y_1=="d" or Y_1=="D":
            Y_1_1yanlis=0
            while Y_1_1yanlis==0:
                Y_1_1=input("Palangada kuvvet kazancı yoktur.")
                print()
                if Y_1_1=="d" or Y_1_1=="D":
                    print("5 Nolu Çıkış...")
                    Y_1_1yanlis=1
                    break
                elif Y_1_1=="y" or Y_1_1=="Y":
                    print("6 Nolu Çıkış...")
                    Y_1_1yanlis=1
                    break
            else:
                print("D veya Y harfleri ile giriş yapınız_!")
                print()
                Y_1_1yanlis=0
        Y_1yanlis=1
    break

elif Y_1=="y" or Y_1=="Y":
    Y_1_2yanlis=0
    while Y_1_2yanlis==0:
        Y_1_2=input("Cisimler iki tür kinetik enerjiye sahip olabilir.")
        print()
        if Y_1_2=="d" or Y_1_2=="D":
            print("7 Nolu Çıkış...")
            Y_1_2yanlis=1
            break
        elif Y_1_2=="y" or Y_1_2=="Y":
            print("8 Nolu Çıkış...")
            Y_1_2yanlis=1
            break
    else:
        print("D veya Y harfleri ile giriş yapınız_!")
        print()
        Y_1_2yanlis=0
    Y_1yanlis=1
    break
else:
    print("D veya Y harfleri ile giriş yapınız_!")
    print()
```

```
Y_1yanlis=0  
Soruyanlis=1  
break
```

```
else:  
print("D veya Y harfleri ile giriş yapınız_!")  
print()
```