

Problem Seti:1

Bu problem seti size Python'da fonksiyonlar yaratma konusunu ve ayrıca bir koşula ulaşılan kadar bir hesaplama sürecini tekrarlamak için döngü mekanizmalarını tanıttacaktır.

İşbirliği

Diğer öğrencilerle birlikte çalışabilirsiniz. Ancak, her öğrenci ödevini ayrı ayrı yazmalı ve teslim etmelidir. *Gönderinizin yorumlarında kimlerle çalıştığınızı belirttiğinizden emin olun.*

Problem 1: Basit Adam Asmaca

Klasik kelime oyunu Adam Asmaca'nın bir varyasyonunu uygulayacaksınız. Oyunun kurallarına aşina değilseniz, [https://en.wikipedia.org/wiki/Hangman_\(game\)](https://en.wikipedia.org/wiki/Hangman_(game)) sayfasını okuyun. Bu problemde korkmayın, aslında görüldüğünden daha kolay! Bu problem için 'darağacı' kuracağız ve asıl oyunu uygulamadan önce yardımcı fonksiyonların oluşturulmasında size rehberlik edeceğim.

A) Hadi Başlayalım

'adamAsmaca.py' ve "kelimeler.txt" dosyalarını indirin ve **ikisini de aynı dizine kaydedin**. Dosyalarınızın doğru şekilde kaydedildiğinden emin olmak için herhangi bir kod yazmadan önce hangman.py dosyasını çalıştırın. Size verdiğimiz kod, bir dosyadan kelimelerle yüklenir. Kabuğunuzda aşağıdaki çıktıyı görmelisiniz:

```
Dosyadan kelime listesi yükleniyor...  
55900 kelime yüklendi.
```

Yukarıdaki metni görürseniz Adam Asmaca Oyunu Gereksinimleri ile devam edin. Bunu yapmazsanız, her iki dosyanın da aynı yere kaydedildiğini iki kez kontrol edin!

B) Adam Asmaca Oyunu Gereksinimleri

Kullanıcının bilgisayara karşı adam asmaca oynamasına izin verecek olan `adamAsmaca` adlı bir fonksiyon uygulayacaksınız. Bilgisayar kelimeyi seçer ve oyuncu kelimedeki harfleri tahmin etmeye çalışır.

İşte uygulamak istediğimiz genel davranış. Gözünüz korkmasın! Bu sadece bir açıklama; **bunu adımlara ayıracağız ve daha sonra problem setinde daha fazla işlevsel özellik sağlayacağız, bu yüzden okumaya devam edin!**

1. Bilgisayar, kelimeler.txt dosyasında sağlanan mevcut kelimeler listesinden rastgele bir kelime seçmelidir.

kelimeler.txt dosyasının tamamı İngilizce küçük harfli kelimeler içerdiğini unutmayın.

2. Kullanıcıya başlangıçta belirli sayıda tahmin verilir.

3. Oyun etkileşimlidir; kullanıcı tahminini ve bilgisayarı şu şekilde girer:

a) Gizli kelimedede varsa harfi ortaya çıkarır veya

b) kullanıcıyı cezalandırır ve kalan tahmin sayısını günceller.

4. Oyun, kullanıcı gizli kelimeyi tahmin ettiğinde veya kullanıcının tahminleri bittiğinde sona erer.

Problem 2:

Adam Asmaca Bölüm 1: Üç yardımcı fonksiyon

Adam asmaca oyununu düzenlemek için kod yazmadan önce, sorunu mantıksal alt görevlere ayıracağız ve bu oyunun çalışması için sahip olmanız gereken üç yardımcı fonksiyon tanımlayacağız. Bu, hesaplamalı problem çözmeye yönelik yaygın bir yaklaşımdır ve sizin deneyimlemeye başlamanızı istediğim bir yaklaşımdır.

`adamAsmaca.py` dosyası, çözümünüzü yazarken kullanabileceğiniz, önceden uygulanmış bir dizi işleve sahiptir. Her bir yardımcı işlevin nasıl kullanılacağını belge dizilerini okuyarak anlamanız gerekse de, dosyanın üst kısmındaki sizin için zaten uygulanmış olan iki işlevdeki kodu yok sayabilirsiniz.

1A) Kelimenin tahmin edilip edilmediğini belirleyin

İlk olarak, bir string, `secret_word` ve bir harf (dize) listesi, `letters_guessed` olmak üzere iki parametre alan `is_word_guessed` fonksiyonunu uygulayın. Bu fonksiyon, `secret_word` tahmin edildiye (yani, tümü) bir boole `True` ifadesi döndürür. `secret_word`'ün harfleri `letters_guessed` şeklindedir, aksi takdirde `False` şeklindedir. Bu fonksiyon, adam asmaca oyununun ne zaman başarıyla tamamlandığına karar vermenize yardımcı olacak ve harfleri gizli kelimeye karşı kontrol eden herhangi bir yinelemeli döngü için bir son test haline gelecektir.

Bu fonksiyon için, `secret_word` ve `letter_guessed` içindeki tüm harflerin küçük harf olduğunu varsayabilirsiniz.

Örnek Kullanım:

```
>>> secret_word = 'apple'
>>> letters_guessed = ['e', 'i', 'k', 'p', 'r', 's']
>>> print(is_word_guessed(secret_word, letters_guessed))

False
```

1B) Kullanıcının tahminini almak

Ardından, iki parametre alan `get_guessed_word` fonksiyonunu uygulayın - bir dize, `secret_word` ve bir harf listesi, `letter_guessed`. Bu fonksiyonu, `letter_guessed`'deki harflerin, `secret_word`'deki hangi harflerin olduğuna bağlı olarak, harflerden ve alt çizgilerden oluşan bir dize döndürür. Bu, `is_word_guessed`'den çok farklı olmamalıdır!

Bilinmeyen harfleri temsil etmek için bir alt çizgi ve ardından bir boşluk (_) kullanacağız. Başka semboller seçebildik, ancak alt çizgi ve boşluk kombinasyonu görülebilir ve kolayca ayırt edilebilir. Boşluğun çok önemli olduğuna dikkat edin, aksi takdirde ____'nin dört eleman mı yoksa üç eleman mı olduğunu ayırt etmek zor. Buna *kullanılabilirlik* denir, programlama yaparken programınızın kullanılabilirliğini göz önünde bulundurmak çok önemlidir. Kullanıcılar programınızı anlamakta veya çalıştırmakta zorlanırsa, programı kullanmazlar! Programınızı tasarlarken kullanılabilirliği düşünmenizi öneririz.

İpucu: Fonksiyonunuzu tasarlarken, işiniz bittiğinde hangi bilgileri geri döndürmek istediğinizi, bir veri yapısı üzerinde döngü yaparken bu bilgileri depolamak için bir yere ihtiyacınız olup olmadığını ve birikmiş sonuçlarınıza nasıl bilgi eklemek istediğinizi düşünün.

Örnek Kullanım:

```
>>> secret_word = 'apple'
>>> letters_guessed = ['e', 'i', 'k', 'p', 'r', 's']
>>> print(get_guessed_word(secret_word, letters_guessed))

'_ pp_e'
```

1C) Mevcut tüm harfleri almak

Ardından, bir parametrede harflerden oluşan bir listeyi alan `get_available_letters` fonksiyonunu uygulayın, `letters_guessed`. Bu fonksiyon, İngilizce küçük harflerden oluşan bir dize döndürür - tümü `letter_guessed` içinde olmayan İngilizce küçük harfler.

Bu fonksiyon harfleri alfabetik sırayla döndürmelidir. Bu fonksiyon için, `letter_guessed` içindeki tüm harflerin küçük harf olduğunu varsayabilirsiniz.

İpucu: Tümü küçük harflerden oluşan bir dize olan `string.ascii_lowercase` kullanmayı düşünebilirsiniz:

```
>>> import string
>>> print(string.ascii_lowercase)
abcdefghijklmnopqrstuvwxyz
```

Örnek Kullanım:

```
>>> letters_guessed = ['e', 'i', 'k', 'p', 'r', 's']
>>> print get_available_letters(letters_guessed)
abcdefghijklmnopqrstuvwxyz
```

Problem 3:

Adam Asmaca Bölüm 2: Oyun

Artık bazı yararlı fonksiyonlar oluşturduğunuza göre, kullanıcının tahmin etmesi gereken `secret_word`'ün bir parametresini alan `adamAsmaca` fonksiyonunu uygulamaya dönebilirsiniz. Başlangıçta, bu fonksiyonu çalıştırdığınızda bu gizli kelimeyi manuel olarak ayarlayabilirsiniz (ve bunu yapmalısınız da!) – bu, kodunuzu test etmeyi kolaylaştıracaktır. Ama sonunda, sizi veya başka bir kullanıcıyı bu işlevi çalıştırarak oyunu oynamaya davet etmeden önce bilgisayarın bu gizli kelimeyi rastgele seçmesini isteyeceksiniz.

`adamAsmaca` fonksiyonunun çağırılması, kullanıcı ve bilgisayar arasında etkileşimli bir `adamAsmaca` oyununu başlatır. Kodunuzu tasarlarken, önceki bölümde tanımladığınız `is_word_guessed`, `get_guessed_word` ve `get_available_letters` adlı üç yardımcı fonksiyondan yararlandığınızdan emin olun!

Aşağıda, farklı kategorilere ayrılmış oyun gereksinimleri bulunmaktadır. Uygulamanızın tüm gereksinimlere uyduğundan emin olun!

OYUN GEREKSİNİMLERİ

A. Oyun Mimarisi:

1. Bilgisayar, `kelimeler.txt` dosyasında sağlanan mevcut kelimeler listesinden rastgele bir kelime seçmelidir. Kelime listesini yükleme ve rastgele bir kelime seçme fonksiyonları `adamAsmaca.py`'de zaten sizin için sağlanmıştır.
2. Kullanıcılar 6 tahminle başlar.
3. Oyunun başında, kullanıcıya bilgisayarın kelimesinde kaç harf olduğunu ve kaç tahminle başladığını bildirin.
4. Bilgisayar, kullanıcının o ana kadar tahmin etmediği tüm harfleri takip eder ve her dönüşten önce kullanıcıya “*kalan harfleri*” gösterir.

Örnek Oyun Uygulaması:

```
Dosyadan kelime listesi yükleniyor...
55900 kelime yüklendi.
```

Adam Asmaca oyununa hoş geldiniz!
4 harf uzunluğunda bir kelime düşünüyorum.

6 tahmininiz kaldı.
Kullanılabilir harfler: abcdefghijklmnopqrstuvwxyz

B. Kullanıcı-Bilgisayar Etkileşimi:

Oyun etkileşimli olmalı ve aşağıdaki gibi akmalıdır:

1. Her tahminden önce kullanıcıya şunları göstermelisiniz:
 - a) Kullanıcıya her tahminden sonra kaç tahmin kaldığını hatırlatın.
 - b) kullanıcının henüz tahmin etmediği tüm harfler
2. Kullanıcıdan her seferinde bir tahmin girmesini isteyin. (Kullanıcıdan ne tür girdiler bekleyebileceğini görmek için aşağıdaki kullanıcı girdi gereksinimlerine bakın).
3. Her tahminden hemen sonra kullanıcıya harfin bilgisayar kelimesinde olup olmadığı söylenmelidir.
4. Her tahminden sonra, kullanıcıya bilgisayarın kelimesini, tahmin edilen harfler görüntülenecek ve tahmin edilmeyen harfler bir alt çizgi ve boşluk (_) ile değiştirilerek göstermelisiniz.
5. Tahminin sonunda, bireysel tahminleri birbirinden ayırmaya yardımcı olmak için birkaç tire (— — —) yazdırın.

Örnek Oyun Uygulaması:

(Aşağıdaki **mavi** renk, bilgisayarın çıktısının aksine, yalnızca kullanıcının ne yazdığını size göstermek için vardır.)

```
6 tahmininiz kaldı.  
Kullanılabilir harfler: abcdefghijklmnopqrstuvwxyz  
Lütfen bir harf tahmin edin: a  
İyi tahmin: _ a _ _  
-----  
6 tahmininiz kaldı.  
Kullanılabilir harfler: bcdefghijklmnopqrstuvwxyz  
Lütfen bir harf tahmin edin: b  
Hata! O harf bu kelimeye yok: _ a _ _
```

C. Kullanıcı Giriş Gereksinimleri:

1. Kullanıcının bir seferde yalnızca bir karakter tahmin edeceğini varsayabilirsiniz, ancak kullanıcı herhangi bir sayı, sembol veya harf seçebilir. Kodunuz büyük ve küçük harfleri geçerli tahminler olarak kabul etmelidir!
2. Kullanıcı alfabe (semboller, sayılar) dışında herhangi bir şey girerse, kullanıcıya yalnızca alfabe girebileceğini söyleyin. Kullanıcı bunu yanlışlıkla yapabileceğinden oyunun başında 3 uyarı alması gerekir. Geçersiz bir giriş veya önceden tahmin ettikleri bir harf girdiklerinde, bir uyarıyı kaybetmeleri gerekir. Kullanıcının hiçbir uyarısı kalmamışsa ve geçersiz bir giriş girerse, tahminini kaybetmesi gerekir.

İpucu #1: Kullanıcının tahminini almak için giriş fonksiyonuna yapılan çağrılarını kullanın.

- a) Kullanıcı girişinin bir alfabe olduğunu kontrol edin.
- b) Kullanıcı bir büyük veya küçük harf girmezse, bir uyarı veya bir tahmin çıkarın.

İpucu #2: `str.isalpha('your string')` ve `str.lower('Your String')` fonksiyonlarını faydalı bulabilirsiniz! Bu fonksiyonların ne olduğunu bilmiyorsanız, fonksiyonunlar hakkında detaylı bilgi için Spyder kabuğunuza `help(str.isalpha)` veya `help(str.lower)` yazmayı deneyebilirsiniz.

İpucu #3: kelimeler.txt dosyasındaki kelimeler küçük harfli olduğundan, kullanıcı girdisini her zaman küçük harfe dönüştürmek ve oyununuzun yalnızca küçük harf işlemlerini sağlamak daha kolay olabilir.

Örnek Oyun Uygulaması:

```
3 uyarınız kaldı.  
6 tahmininiz kaldı.  
Kullanılabilir harfler: bcdefghijklmnopqrstuvwxyz  
Lütfen bir harf tahmin edin: s  
Hata! O harf bu kelimede yok: _ a _ _  
-----  
5 tahmininiz kaldı.  
Kullanılabilir harfler: bcdefghijklmnopqrtuvwxyz  
Lütfen bir harf tahmin edin: ş  
Hata! Bu geçerli bir harf değil. 2 uyarınız kaldı: _ a _ _
```

D. Oyunun Kuralları:

1. Kullanıcı 3 uyarı ile başlar.
2. Kullanıcı alfabe (semboller, sayılar) dışında herhangi bir şey girerse, kullanıcıya yalnızca alfabe girebileceğini söyleyin.
 - a) Kullanıcının bir veya daha fazla uyarısı kalmışsa, kullanıcı bir uyarıyı kaybetmelidir. Kullanıcıya kalan uyarı sayısını söyleyin.
 - b) Kullanıcının kalan uyarısı yoksa, bir tahminini kaybetmesi gerekir.
3. Kullanıcı önceden tahmin edilmiş bir harf girerse, kullanıcıya harfin önceden tahmin edildiğini bildiren bir mesaj yazdırın.
 - a) Kullanıcının bir veya daha fazla uyarısı kalmışsa, kullanıcı bir uyarıyı kaybetmelidir. Kullanıcıya kalan uyarı sayısını söyleyin.
 - b) Kullanıcının herhangi bir uyarısı yoksa, bir tahminini kaybetmesi gerekir.
4. Kullanıcı daha önce tahmin edilmemiş bir harf girerse ve harf gizli kelimedeysse, kullanıcı tahmin **kaybetmez**.
5. **Ünsüzler:** Kullanıcı tahmin edilmemiş bir ünsüz girerse ve ünsüz gizli kelimede değilse, ünsüz ise kullanıcı **bir** tahmin kaybeder.
6. **Ünlüler:** Ünlü tahmin edilmemişse ve sesli harf gizli kelimede değilse, kullanıcı **iki** tahmin kaybeder. Ünlüler a, e, i, o ve u'dur. y sesli harf olarak sayılmaz.

Örnek Uygulama:

```
5 tahmininiz kaldı.  
Kullanılabilir harfler: bcdefghijklmnopqrtuvwxyz  
Lütfen bir harf tahmin edin: t  
İyi tahmin: ta_ t  
-----  
5 tahmininiz kaldı.  
Kullanılabilir harfler: bcdefghijklmnopqrtuvwxyz  
Lütfen bir harf tahmin edin: e  
Hata! O harf bu kelimede yok: ta_ t  
-----  
3 tahmininiz kaldı.  
Kullanılabilir harfler: bcdefghijklmnopqrtuvwxyz  
Lütfen bir harf tahmin edin: e  
Hata! Bu harfi zaten tahmin ettin. Artık 2 uyarınız var: ta_ t
```

E. Oyunu Sonlandırma:

1. Oyun, kullanıcı tam kelimeyi oluşturduğunda veya tahminleri bittiğinde sona ermelidir.
2. Oyuncunun kelimeyi tamamlamadan önce tahminleri biterse, onlara kaybettiklerini söyleyin ve oyun bittiğinde kelimeyi kullanıcıya gösterin.
3. Kullanıcı kazanırsa, bir tebrik mesajı yazdırın ve kullanıcıya puanını söyleyin.

4. Toplam puan, kullanıcı gizli_kelimeyi (secret_word'ü), gizli_kelimedeki (secret_word'teki) benzersiz harflerin sayısını tahmin ettiğinde kalan tahmin (guesses_remaining) sayısıdır.

Toplam puan = kalan_tahminler(guesses_remaining) * gizli_kelimedeki benzersiz harflerin sayısı(number unique letters in secret_word)

Örnek Uygulama:

```
3 tahmininiz kaldı.  
Mevcut harfler: bcd fgh i j k l n o p q u v w x y z  
Lütfen bir harf tahmin edin: c  
İyi tahmin: tact  
-----  
Tebrikler kazandın!  
Bu oyun için toplam puanınız: 9
```

Örnek Uygulama:

```
3 tahmininiz kaldı.  
Mevcut harfler: bcd fgh i j k l n o p q u v w x y z  
Lütfen bir harf tahmin edin: n  
İyi tahmin: dolphin  
-----  
Tebrikler kazandın!  
Bu oyun için toplam puanınız: 21
```

F. Genel İpuçları:

1. Gerekirse ek yardımcı işlevler yazmayı düşünün.
2. Saklamak isteyebileceğiniz dört önemli bilgi parçası vardır:
 - a) secret_word: Tahmin edilecek kelime. Bu, adamAsmaca fonksiyonu için parametre adı olarak zaten kullanılıyor.
 - b) letter_guessed: Şimdiye kadar tahmin edilen harfler. Zaten letter_guessed' de olan bir harfi tahmin ederlerse, bunu zaten tahmin ettiklerini söyleyen bir mesaj yazdırmalısınız, ancak bunun için onları cezalandırmayın.
 - c) guesses_remaining: Kullanıcının bıraktığı tahmin sayısı. Örnek oyunumuzda yanlış bir sesli harf seçmenin cezasının yanlış bir ünsüz seçmenin cezasından farklı olduğunu unutmayın.
 - d) warnings_remaining: Kullanıcının bıraktığı uyarı sayısı. Kullanıcının yalnızca önceden tahmin edilmiş bir sembol veya harf girdiğinde bir uyarıyı kaybettiğini unutmayın.

G. Örnek Oyun:

Yukarıda verilen adamAsmaca örneklerine dikkatlice bakın, çünkü bu, her harf tahmininden sonra yazdırmak isteyeceğiniz bilgi örneklerini önerir.

Not: Ekrana yazılan ifadelerinizi mümkün olduğunca örnek oyuna yakın hale getirmeye çalışın!

Kazanan bir oyunun çıktısı şöyle görünmelidir. (Aşağıdaki **mavi renk**, bilgisayarın çıktısının aksine, yalnızca kullanıcının ne yazdığını size göstermek için vardır.)

```
Dosyadan kelime listesi yükleniyor...  
55900 kelime yüklendi.  
Adam Asmaca oyununa hoş geldiniz!  
4 harf uzunluğunda bir kelime düşünüyorum.  
3 uyarınız kaldı.  
-----  
6 tahmininiz kaldı.  
Kullanılabilir harfler: abcdefghijklmnopqrstuvwxyz  
Lütfen bir harf tahmin edin: a  
İyi tahmin: _ a _ _
```

```
-----
6 tahmininiz kaldı.
Kullanılabilir harfler: bcdefghijklmnopqrstuvwxyz
Lütfen bir harf tahmin edin: a
Hata! Bu harfi zaten tahmin ettin. 2 uyarınız kaldı:
_ a _
-----
6 tahmininiz kaldı.
Kullanılabilir harfler: bcdefghijklmnopqrstuvwxyz
Lütfen bir harf tahmin edin: s
Hata! O harf bu kelimede yok.
Lütfen bir harf tahmin edin: _ a _
-----
5 tahmininiz kaldı.
Kullanılabilir harfler: bcdefghijklmnopqrtuvwxyz
Lütfen bir harf tahmin edin: $
Hata! Bu geçerli bir harf değil. 1 uyarınız kaldı: _ a _
-----
5 tahmininiz kaldı.
Kullanılabilir harfler: bcdefghijklmnopqrtuvwxyz
Lütfen bir harf tahmin edin: t
İyi tahmin: ta_ t
-----
5 tahmininiz kaldı.
Kullanılabilir harfler: bcdefghijklmnopqrtuvwxyz
Lütfen bir harf tahmin edin: e
Hata! O harf bu kelimede yok: ta_ t
-----
3 tahmininiz kaldı.
Kullanılabilir harfler: bcd fghijklmnopqrtuvwxyz
Lütfen bir harf tahmin edin: e
Hata! Bu harfi zaten tahmin ettin. 0 uyarınız kaldı: ta_ t
-----
3 tahmininiz kaldı.
Kullanılabilir harfler: bcd fghijklmnopqrtuvwxyz
Lütfen bir harf tahmin edin: e
Hata! Bu harfi zaten tahmin ettin. Hiçbir uyarınız kalmadığından
bir tahmininizi kaybedersiniz: ta_ t
-----
2 tahmininiz kaldı.
Mevcut harfler: bcd fghijkl n o p q u v w x y z
Lütfen bir harf tahmin edin: c
İyi tahmin: tact
-----
Tebrikler kazandın!
Bu oyun için toplam puanınız: 6
```

Ve kaybeden bir oyunun çıktısı şöyle görünmelidir...

```
Dosyadan kelime listesi yükleniyor...
55900 kelime yüklendi.
Adam Asmaca oyununa hoş geldiniz!
4 harf uzunluğunda bir kelime düşünüyorum
3 uyarınız kaldı.
-----
6 tahminin kaldı
Mevcut Harfler: abcdefghijklmnopqrstuvwxyz
Lütfen bir harf tahmin edin: a
Hata! O harf bu kelimede yok: _ _ _ _
-----
```

```
4 tahminin kaldı
Mevcut Harfler: bcdefghijklmnopqrstuvwxyz
Lütfen bir harf tahmin edin: b
Hata! O harf bu kelimede yok: _ _ _ _
-----
3 tahminin kaldı
Mevcut Harfler: cdefghijklmnopqrstuvwxyz
Lütfen bir harf tahmin edin: c
Hata! O harf bu kelimede yok: _ _ _ _
-----
2 tahminin kaldı
Mevcut Harfler: defghijklmnopqrstuvwxyz
Lütfen bir harf tahmin edin: 2
Hata! Bu geçerli bir mektup değil. 2 uyarınız kaldı: _ _ _ _
2 tahminin kaldı
Mevcut Harfler: defghijklmnopqrstuvwxyz
Lütfen bir harf tahmin edin: d
Hata! O harf bu kelimede yok: _ _ _ _
-----
1 tahmininiz kaldı
Mevcut Harfler: efghijklmnopqrstuvwxyz
Lütfen bir harf tahmin edin: e
İyi tahmin: e _ _ e
-----
1 tahmininiz kaldı
Mevcut Harfler: fghijklmnopqrstuvwxyz
Lütfen bir harf tahmin edin: f
Hata! O harf bu kelimede yok: e _ _ e
-----
Üzgünüm, tahminleriniz tükendi. Kelime başkaydı.
```

Kodunuzu tamamlayıp test ettikten sonra (“gizli” kelimeyi manuel olarak sağladınız, çünkü bunun kodunuzda hata ayıklamanıza yardımcı olduğunu biliyorsunuz), bilgisayara karşı çalıştırmayı denemek isteyebilirsiniz. Sağladığımız dosyanın en altına inerseniz, metnin altında iki yorum satırı göreceksiniz.

```
if __name__ == "__main__":
    #secret_word = choose_word(kelime_listesi)
    #adamAsmaca(secret_word)
```

Bu satırlar, incelemek isteyebileceğiniz, size sağlanan (adamAsmaca.py'nin üst kısmına yakın) fonksiyonları kullanır. Bu satırları “# işaretini kaldırmayı” ve kodunuzu yeniden yüklemeyi deneyin. Bu size, büyük bir sözcük kümesi yüklemek ve ardından rastgele bir sözcük seçmek için fonksiyonlarımızı kullanan bilgisayara karşı becerinizi denemeniz için bir şans verecektir.

Problem 4:

Adam Asmaca Bölüm 3: İpuçları Olan Oyun

Adam Asmaca'yı bilgisayara karşı oynamayı denediyseniz, özellikle “esoteric” bir kelime (“esoteric” gibi!) seçtiğinde, bilgisayarı yenmenin her zaman kolay olmadığını fark etmişsinizdir. Bilgisayardan şu anda tahmin ettiğinizle eşleşen tüm kelimelerin bir listesi gibi bir ipucu istemeniz iyi olabilir.

Örneğin, gizli kelime “tact” ise ve şimdiye kadar “t” harfini tahmin ettiyseniz, çözümün “t _ t” olduğunu bilirsiniz, burada iki eksik harfi tahmin etmeniz gerekir. Eşleşen sözcük kümesinin (en azından bilgisayarın başlangıçta ne yüklediğine bağlı olarak) olduğunu bilmek güzel olabilir:

```
tact tart taut teat tent test text that tilt tint toot tort tout
trot tuft twit
```


Özel karakter '*'ı tahmin ederseniz bilgisayarın yüklenen tüm kelimeleri bulacağı özelliğiyle, Adam Asmaca'nın bir varyasyonunu (biz buna adamAsmaca_ipuclu diyoruz ve bunu yazmak için bir başlangıç darağacı sağladık) yaratmanızı sağlayacağız. Mevcut tahmin ettiğiniz kelimeyle eşleşebilecek bir liste yapın ve her birinin çıktısını alın. Tabii ki, bunu ilk adımda denemenizi önermiyoruz, çünkü bu yüklediğimiz 55.900 kelimenin tamamını yazdıracaktır! Ancak bir cevaba yaklaşıyorsanız ve tahminleriniz tükeniyorsa, bu yardımcı olabilir. Bunu yapmak için, önce iki yardımcı fonksiyonu tamamlamanızı isteyeceğiz:

3A) Mevcut tahmin edilen kelimeyi eşleştirme

`match_with_gaps` iki parametre alır: `my_word` ve `other_word`. `my_word` tahmini bir kelimenin bir örneğidir, başka bir deyişle, bazı yerlerde _ 'ler olabilir ('t_ _ t' gibi). `other_word` normal bir İngilizce kelimedir.

`my_word`'ün tahmin edilen harfleri, `other_word`'ün karşılık gelen harfleriyle eşleşiyorsa, bu fonksiyon `True` döndürmelidir. İki kelime aynı uzunlukta değilse veya `my_word`'deki tahmin edilen bir harf `other_word`'teki karşılık gelen karakterle eşleşmiyorsa `False` döndürmelidir.

Bir harf tahmin edildiğinde, kodunuz o harfin gizli kelimede geçtiği tüm konumları ortaya çıkardığını unutmayın. Bu nedenle, gizli harf (_) sözcüğünde daha önce ortaya çıkan harflerden biri olamaz.

Örnek Kullanım:

```
>>> match_with_gaps("te_ t", "tact")
False
>>> match_with_gaps("a_ _ le", "banana")
False
>>> match_with_gaps("a_ _ le", "apple")
True
>>> match_with_gaps("a_ ple", "apple")
False
```

İpucu: Uzunlukları karşılaştırırken sözcükteki boşluklardan kurtulmak için `strip()` kullanmak isteyebilirsiniz.

3B) Tüm olası eşleşmeleri gösterme

`show_possible_matches` tek bir parametre alır: `my_word` bu, tahmin edilen bir kelimenin bir örneğidir, başka bir deyişle, bazı yerlerde _ 'ler olabilir ('t_ _ t' gibi).

Bu fonksiyon, sözcük listesindeki tüm sözcükleri yazdırmalıdır (bunu dosyanın başında, satır 51'de tanımladığımıza dikkat edin) `my_word` ile eşleşen. Eşleşme yoksa "Eşleşme bulunamadı" yazmalıdır.

Örnek Kullanım:

```
>>> show_possible_matches("t_ _ t")
tact tart taut teat tent test text that tilt tint toot tort tout
trot tuft twit
>>> show_possible_matches("abbbb_")
Hiçbir sonuç bulunamadı
>>> show_possible_matches("a_ pl_ ")
ample amply
```

3C) Hangman with hints

Şimdi, `adamAsmaca` için yazdığınız kodu `adamAsmaca_ipuclu`'in gövdesi olarak çoğaltabilmeli, ardından kullanıcının bir yıldız işareti (*) tahmin edebileceği duruma izin vermek için küçük bir

ekleme yapabilmelisiniz. Bu durumda bilgisayar bu tahminle eşleşen tüm kelimeleri yazdıracaktır.

Tahmin bir yıldız ise, kullanıcı tahminini kaybetmemelidir.

Orijinal adamAsmaca oyununu oynamak için kullandığınız kod satırlarındaki # işaretini silerek satırları aktif edin:

```
secret_word = choose_word(wordlist)
adamAsmaca(secret_word)
```

İpuçları ile yeni adamAsmaca'yı oyununuzu oynamak için dosyanın alt kısmında sağladığımız bu kod satırlarındaki # işaretini silerek satırları aktif edin:

```
#secret_word = choose_word(wordlist)
#adamAsmaca_ipuclu(secret_word)
```

Örnek Çıktı:

Yıldız işareti tahmininden elde edilen çıktı, aşağıdaki örnek çıktıya benzemelidir. Diğer tüm çıktılar, yukarıdaki Bölüm 2'de açıklanan adamAsmaca oyununu izlemelidir.

```
Dosyadan kelime listesi yükleniyor...
      55900 kelime yüklendi.
Adam Asmaca oyununa hoş geldiniz!
5 harf uzunluğunda bir kelime düşünüyorum.
-----
6 tahmininiz kaldı.
Kullanılabilir harfler: abcdefghijklmnopqrstuvwxyz
Lütfen bir harf tahmin edin: a
İyi tahmin: a _ _ _ _
-----
6 tahmininiz kaldı.
Kullanılabilir harfler: bcdefghijklmnopqrstuvwxyz
Lütfen bir harf tahmin edin: l
İyi tahmin: a _ _ l _
-----
6 tahmininiz kaldı.
Kullanılabilir harfler: bcdefghijklmnopqrstuvwxyz
Lütfen bir harf tahmin edin: *
Olası kelime eşleşmeleri şunlardır:
addle adult agile aisle amble ample amply amyls angle ankle apple
apply aptly arils atilt
-----
6 tahmininiz kaldı.
Kullanılabilir harfler: bcdefghijklmnopqrstuvwxyz
Lütfen bir harf tahmin edin: e
İyi tahmin: a _ _ le
-----
```

Bu, problem setini tamamlar!