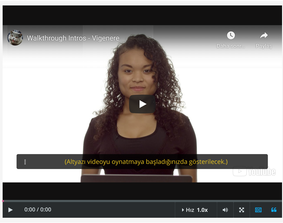
"Vigenere" Görevi Tanımı

Bu sayfaya yer imi koy

Video



Altyazının başlangıcı. Sona atla.

Diyelim ki arkadaşınıza yeni bir mesaj ileteceksiniz.
Sezar şifresinden daha güvenli bir çözüme ihtiyacınız var.
O zaman Vigenere şifresini seçebilirsiniz!
Vigenere, basit bir anahtar yerine daha karmaşık bir anahtar kelime kullanır.
Bu anahtar kelimenin her harfi bir şifreyi temsil eder.
A harfi, 0 kaydırma demektir.
Çünkü A ilk harftir, ve saymaya sıfırdan başlarız.
B harfi ise 1 kaydırmayı temsil eder.

Öncelikle CS50 Sandbox'ta bu sayfayı açın ve aşağıdakileri takip edin:

Ooh, la la!

Vigenère'in şifresi, mesajları bir dizi anahtarla şifreleyerek (veya başka bir deyişle, bir anahtar kelimeyi kullanarak) Sezar'ın şifresini geliştirir.

Başka bir deyişle, p bir düz metinse ve k bir anahtar kelimeyse (yani alfabetik bir string ise, burada A (veya a) 0, B (veya b) 1, C (veya c) 2,... ve Z (veya z) 25'i temsil eder), o zaman şifre metin, c, içinde bulunan her bir harf, c[i], şöyle hesaplanır:

c[i] = (p[i] + k[i]) % 26

Bu şifrenin sadece k yerine k[i] kullanımına dikkat edin. Ve k, p'den daha kısaysa, k içindeki harfler p'yi şifrelemek için döngüsel olarak tekrar tekrar kullanılmalıdır.

Başka bir deyişle eğer Vigenère birine gizlice HELLO demek isterse ve ABC anahtar kelimesiyle, bunu şu şekilde yapabilir: H'yi 0 anahtarıyla (yani A), E'yi 1 anahtarıylayani B), ilk L'yi 2 anahtarıyla (yani C) şifreler, bu noktadan sonra elindeki anahtar kelimenin karakterleri bitt@ için tekrardan anahtar kelimenin başına dönecek ve ikinci L harfini 0 anahtarıyla (yani A) ve O'yu da 1 anahtarıyla (yani B) şifreleyecek. Ve böylece HELLO'yu HFNLP olarak şifyle yazacaktır:

düzmetin	H	E	L	L	O
+ anahtar	A	B	C	A	B
(kayma değeri)	0	1	2	0	1
= şifremetin	H	F	N	L	P

Şimdi Vigenère'in şifresini kullanarak mesajları şifrelemenizi sağlayan vigenere adlı bir program yazalım. Kullanıcı programı yürütürken, bir komut satırı argümanı sağlayarak, programın döndürece@i gizli mesajı için anahtar kelimenin ne olması gerektiğine karar vermelidir.

İşte programın nasıl çalışabilece@ine dair birkaç örnek:

```
$ ./vigenere bacon
plaintext:  Meet me at the park at eleven am
ciphertext: Negh zf av huf pcfx bt gzrwep oz
```

veya kullanıcı tam olarak alfabetik olmayan bir anahtar kelime sağladıysa:

```
$ ./vigenere 13
Usage: ./vigenere keyword
```

veya hiç anahtar kelime sağlamadıysa:

```
$ ./vigenere
Usage: ./vigenere keyword
```

veya çok fazla anahtar kelime sağladıysa:

```
$ ./vigenere bacon and eggs
Usage: ./vigenere keyword
```

Deneyin

Ekibimizin bu problem için yazdığı çözümü denemek için, bu sandbox içinde anahtar kelime (keyword) yerine geçerli bir alfabetik doğe kullanın.

./vigenere keyword

Nasıl başlayalım? Tamdık bir şeyle!

Déjà vu

Daha önce de kavradığınız üzere, bu şifrenin temel mantığı Sezar'ın şifresinin altında yatan mantığa çarpıcı bir şekilde benzemektedir. Bu nedenle, Sezar kodumuz başlamak için iyi bir yer gibi görünüyor, bu yüzden "vigenere.c" isim tüm içeriğini "caesar.c" çözümünde değiştirerek başlamaktan çekinmeyin.

Sezar ve Vigenère şifreleri arasındaki bir fark, Vigenère'in anahtarının sayıdan ziyade bir dizi harf olmasıdır. Yani kullanıcının bize bir anahtar kelime verdiğinden emin olalım! Anahtar kelimenin her karakterinin bir rakam yerine alfabetik olduğundan emin olmak için Sezar'da uyguladığınız kontrolü değiştirin. Bunlardan herhangi biri değilse, Usage: ./vigenere keyword yazdırın ve daha önce yaptığımız gibi sıfırdan farklı bir değer döndürün. Hepsi alfabetik ise, kontrol ettikten sonra Success ve, return 0; yazdırmalısınız (şimdilik), zira şifreleme kodumuz henüz tam olarak çalışmaya hazır olmadığından çalıştırmayacağız.

Örnek davranış:

```
$ ./vigenere alpha
Success
```

ya da

```
$ ./vigenere 123
Usage: ./vigenere keyword
```

İpuçları

- string.h başlık dosyasının, string'lerle çalışan bir dizi yararlı fonksiyon içerdiğini unutmayın. Bunlardan bazıları için CS50 Referans menüsüne bakın!
- Uzunluğunu biliyorsanız, bir string'in her bir karakterini yinelemek için döngü kullanabileceğimizi hatırlayın.
- Ctype.h başlık dosyasının, karakterler hakkında bize bilgi veren bir dizi yararlı fonksiyon içerdiğini hatırlayın. Bunlardan bazıları için CS50 Referans menüsüne bakın!

Kaydırma değerini alma

Şimdilik kullanıcının tek karakterlik anahtar kelimeler sağladığını varsayalım. Bu karakteri doğru kaydırma değerine dönüştürebilir miyiz? Bunu bir fonksiyon yazarak yapalım.

Dosyanızın üst kısmına yakın, #include satırlarının altında, amacı tam olarak bunu yapmak olan yeni bir fonksiyon için prototip tanımlayalım. Girdi olarak tek bir karakter alacak ve bu karakter için kaydırma değerini verecektir.

```
int shift(char c);
```

Şimdi, giriş olarak tek bir karakter(s) alan ve bir tamsayı çıkaran shift adlı bir fonksiyon tanımladık.

Şimdi, main bloğunun altında, kendimize bu yeni fonksiyonu tanımlamak (yani uygulamak) için bir yer verelim.

```
int shift(char c)
{
    // TODO
}
```

TODO yazan yerde karakteri konumsal tam sayı değerine dönüştürme işini yapacağız (yani, yine A veya a 0, B veya b 1, Z veya z 25, vb. olacaktır)

Bunu test etmek için, "Success" yazdığınız satırı silin (ancak şimdilik "return 0"ı bırakın) ve silinen satır yerine, kodunuzun çalışıp çalışmadığını test etmek için aşağıdaki satırları ekleyin.

```
int key = shift(argv[1][0]);
printf("%i\n", key);
```

Programınız A veya a anahtar sözcüğüyle çalıştırılırsa 0 yazmalıdır. Anahtar kelime olarak programı diğer büyük ve küçük harflerle çalıştırmayı deneyin. Davranış bekledi@iniz gibi mi?

İpuçları

- Fonksiyonların girdileri ve çıktıları vardır.
- Bir fonksiyon tanımladığımızda, her birinin de bir tipi olan, dönüş tipini, adını ve argüman listesinide sağlamamız gerekiyor.
- Bir fonksiyonu kullandığımızda veya çağırdığımızda, yalnızca argüman listesine uygun değerleri ekleriz ve fonksiyonun çıktısını fonksiyonun dönüş tipine karşılık gelen bir değişkene atarız.
- Argv [1] bir string ise, argv [1] [0] bu string'in sadece ilk karakteridir.
- Ctype.h başlık dosyasının, karakterler hakkında bize bilgi veren bir dizi yararlı fonksiyon içerdiğini hatırlayın.
- A'nın ASCII değeri 65'tir. a'nın ASCII değeri 97'dir.
- B'nin ASCII değeri 66'dır. b'nin ASCII değeri 98'dir. Ortaya çıkan bir potansiyel örüntüyü görüyor musunuz?

Tek karakterlik anahtar kelimeler

Daha önce yazdığınız şifreleme kodunu kullanmaya başlama zamanı! K anahtar kelimeniz tam olarak bir harf içeriyorsa (örneğin, H veya h), Vigenère'in şifresinin etkili bir şekilde bir Sezar şifresi haline geldiğini fark etmiş olabilirsiniz (bu 7. örnekte). Şimdilik gerçekten kullanıcının anahtar kelimesinin tek bir harf olacağını varsayalım. Sağlanan harfin kaydırma değerini hesaplamak için yeni yazdığınız "shift" fonksiyonunu kullanın, bu fonksiyonun dönüş değerini tam sayı değişkeni "anahtara"a atayın ve "anahtarı"ı tam olarak Sezar'ın şifresinde yaptığınız gibi kullanın! Aslında şimdi yeni eklediğiniz printf ve return 0; satırlarını silmeniz, programın daha önce yazdığınız Sezar şifreleme kodu gibi çalışması için yeterli olacaktır!

```
$ ./vigenere A
plaintext:  hello
ciphertext: hello
```

ya da

```
$ ./vigenere b
plaintext:  HELLO
ciphertext: IFMMP
```

ya da

```
$ ./vigenere C
plaintext:  HeLlO
ciphertext: JgNnQ
```

İpuçları

Sezar çözümünüzdeki bazı değişkenleriniz bu laboratuvarda şu ana kadar adlandırılanlarla eşleşmiyorsa, eşleşmeleri için adlarını düzenleyin!

Son Adımlar

Şimdi "vigenere.c" de kalan işlevselliği yazarak işleri bitiş çizgisine götürme sırası sizde. Kullanıcının anahtar kelimesinin muhtemelen birden çok harften oluşacağını unutmayın, bu nedenle düz metnin her harfi için yeni bir kaydırma değeri hesaplamanız gerekebilir; bunu yaptıktan sonra "shift" fonksiyonunuzu dönüşünüzde kullanmak isteyebilirsiniz.

Ayrıca, bir karakteri her şifrelediğinizde, anahtar kelime olan k'deki bir sonraki harfe, (ve tüm karakterleriniz tüketirseniz anahtar kelimenin başına sarılır) ihtiyacınız olduğunu unutmayın. Ancak bir karakteri (ör. Boşluk veya noktalama işareti) şifrelemiyorsanız, bir sonraki k karakterine ilerlemeyin!

Ve daha önce olduğu gibi, büyük/küçük harfleri korumağınızdan emin olun, ama bunu sadece orijinal mesajın durumuna göre yapın. Anahtar kelimedeki bir harfin büyük harfle yazılıp yazılmadığının, şifreli metindeki bir harfin değişip değişmemesi üzerinde hiçbir etkisi olmamalıdır!

İpuçları

- Muhtemelen iki sayaca ihtiyacınız olacak, bir tane düz metin üzerine yineleme yapmak için i; ve bir tane de , anahtar kelime üzerinde yineleme yapmak için j.
- Düz metin üzerinde yineleme yapmak için kullandığınız for döngüsüne güvenmek yerine, anahtar kelime sayacını kendiniz kontrol etmenin kolay bir yolunu bulacaksınız!
- Anahtar kelimenin uzunluğu 4 karakter ise, o anahtar kelimenin son karakteri keyword[3]'te bulunabilir. Ardından, şifrelediğiniz bir sonraki karakter için keyword[0]'ı kullanmak isteyeceksiniz.

Şimdi sizin ödeviniz:

Ödevinizi, bir sonraki ekranda çıkan kutuya kodunuzu yazarak göndereceksiniz. Ödevinizi bize göndermeden, aşağıdaki ödev tanımını okuduğunuzdan emin olun. Sonra da göndlerirken dikkat etmeniz gerekenler şunlar:

- Bu ödevden bir puan alacaksınız ve ilerleyişinize işlenecek.
- Ödevi sistemde bize göndermeden önce, mutlaka CS50 Lab'de test etmenizi öneririz.
- Eğer sonucunuz yanlışsa 0 puan, doğruysa 1 puan alacaksınız. Doğru yapana kadar birkaç kez deneme şansınız var.
- Eğer sonucunuz yanlışsa, aşağıda "See Output" kısmından neler olduğunu inceleyebilirsiniz.
- **Cevabınız doğru olduğunda, mutlaka Gönder tuşuna basıp sonucu bize göndermeyi unutmayın!**
- İnternet bağlantınız ve problemlere bağlı olarak, sonuçların otomatik değerlendirilmesi bazen uzun sürebilir. Bu durumda 5-10 dakika beklemeniz gerekebilir. Eğer çalışmıyorsa sayfanızı yenileyebilirsiniz.