

CENG202 HESAPLAMA TEKNİKLERİ (BİL202) - ÖDEV 2

Son Teslim Tarihi : **11 Mart 2011**

(!) Ödevlerinizi, kendinize ait “@bil.omu.edu.tr” uzantılı elektronik posta adreslerinizden **ceng202hestek@bil.omu.edu.tr** adresine gönderiniz. Konu kısmında 2. ödev olduğunu belirtiniz.

Bölüm 1 : Çarpma & Zincir Yapısı ile Çakışma İdaresi - 100 puan

Ödevin bu bölümünde sizden, 1. ödevinizde tasarlamış olduğunuz **hash** fonksiyonunuzun birden çok sayıda TC kimlik numarası için aynı çarpı değerini üretmesi durumunda derste anlatılan zincir yapısını kullanarak çakışma problemini çözmeniz beklenmektedir. Buna göre:

- Programınızı yazacağınız modülün (betik dosyasının) ismini, **z[öğrenci numaranız]** şeklinde belirleyiniz. Örneğin, numarası 09876543 olan bir öğrencinin hazırlayacağı betik dosyası, “**z09876543.py**” isminde olmalıdır. Numaranızın ilk hanesi “0” olsa da bu “0” ı kullanınız.
- Modülünüzün içerisinde, bir önceki ödevinizde hazırladığınız **hash** fonksiyonu ve bu ödevde özgü olan “**hash_chain**” isimli diğer bir fonksiyon olmalıdır. Aksi halde ödeviniz değerlendirilmeyecektir.
- **hash_chain** fonksiyonunuz argüman almamalıdır. Betik dosyanızın bulunduğu dizine, size 1. ödevde verilmiş olan csv dosyası, “**ogrenci10.csv**” ismi ile koyulacaktır (İsme dikkat ediniz.).
- **hash_chain** fonksiyonunuz, ilgili csv dosyasındaki tüm kimlik numaralarını sırasıyla **hash** fonksiyonuna verecek (1. ödevinizdeki fonksiyonunuzu çağırarak), bu numaralar için birer indeks değeri ürettirecektir. Daha sonra her bir kimlik numarası, 75 elemanlı bir listenin uygun (kendisi için belirlenen çarpı değerinin 1 eksiği, çünkü indeks numaraları 0’ dan başlar) indeksine **string** olarak yazılacaktır. Bir indekse birden çok kimlik numarası denk gelirse, bu indeksteki eleman da bir liste halini alacak ve içerisinde bu indekse yönlendirilmiş olan kimlik numaraları bulunacaktır. **hash_chain fonksiyonunuz, bahsedilen 75 elemanlı listeyi döndürecektir.**
- Sonuç olarak döndüreceğiniz listede toplamda 75 tane kimlik numarası bulunmalıdır. Eğer **hash** fonksiyonunuz çakışmasız çalışmışsa, döndüreceğiniz listenin her bir elemanı birer kimlik numarası olacak ve iç içe listeler bulunmayacaktır. Eğer çakışma olmuş ise, döndüreceğiniz listenin kimi elemanları **None** olurken kimileri de içerisinde birden çok kimlik numarası bulunduran listeler olacaktır.
- Döndüreceğiniz listede ne kadar az **None** değerli eleman varsa puanınız o kadar yüksek olacaktır. Sözelimi, 5 tane **None** değer bulunduran ve çakışma idaresinin hatasız yapıldığı (75

elemanın da ana listede bulunduğu) listeyi üretebilen programın puanı **70** olarak verilecek ve bu bölüm için, 75 puan 100 puana denk gelecek biçimde ölçekleme yapılacaktır.

- **Betik dosyası isimlerine, fonksiyon isimlerine ve csv dosyası isimlerine dikkat ediniz. Aksi durumda ödeviniz değerlendirilmeyecektir.**

Bölüm 2 : Kimlik Numarasından İsim Öğrenme - 25 puan (ekstra)

Ödevin bu bölümünde, “**z09876543.py**” isimli betik dosyanıza “**hash_name**” isimli yeni bir fonksiyon koymanız gerekmektedir. Bu fonksiyon argüman olarak **string** tipinde kimlik numarası almalı ve bu numaraya karşılık gelen isim ve soy isim bilgilerini, **hash** fonksiyonu ile çırpı değeri üreterek ve **hash_chain** fonksiyonundan yararlanarak ekrana yazdırmalıdır. Sadece csv dosyasında arama yaparak ad-soyad bilgilerini yazdıran gerçekleştirimler dikkate alınmayacaktır. Bu nedenle, bir önceki bölümde bahsedilen ve kimlik numaralarının uygun indekslerde tutulduğu 75 elemanlı listeye paralel olarak tutulan ve kimlik numaraları yerine isim bilgilerini içeren ikinci bir listeden yararlanmayı düşünebilirsiniz.