10-Özyinelemeli bir fonksiyon yazarken, yazmanız gereken iki durum nedir?

Temel Durum (Base Case): Özyinelemeli bir fonksiyonun sonsuz bir döngüye girmemesi için bir temel durum belirlenmelidir. Temel durum, fonksiyonun tekrar çağrılmadan önce ulaşması gereken bir durumdur. Bu durum, fonksiyonun işlemini sonlandırır.

Özyinelemeli Durum (Recursive Case): Fonksiyonun her tekrar çağrıldığında orijinal problemin daha küçük bir versiyonunu ele alması gerekmektedir. Bu adımlar, genellikle orijinal problemi parçalayan ve küçülten işlemlerdir. Bu sayede fonksiyon, temel duruma doğru ilerler.

Bu iki durumu dikkate alarak, özyinelemeli bir fonksiyonu doğru ve etkili bir şekilde yazabilirsiniz.

11-Eğer özyinelemeli bir fonksiyonun temel durumu olmasaydı, çağrıldığında ne olurdu?

Eğer bir özyinelemeli fonksiyonun temel durumu bulunmuyorsa, fonksiyon sürekli olarak kendisini tekrar çağırır ve hiçbir koşul altında durmaz. Bu durum, sonsuz bir özyineleme döngüsüne yol açar ve fonksiyonun çalışması hiçbir zaman sonlanmaz. Bilgisayarın belleğinde ve işlemcisinde sınırlı kaynaklar olduğu için, bu tür bir durum, programın yetersiz kaynaklar nedeniyle çökmesine veya hata vermesine neden olabilir.

12- 12-Özyinelemeli bir işlevdeki özyinelemeli çağrı için doğru olması gereken nedir? Başka bir deyişle, bu özyinelemeli çağrıyı yaparken neyi sağlamalısınız?

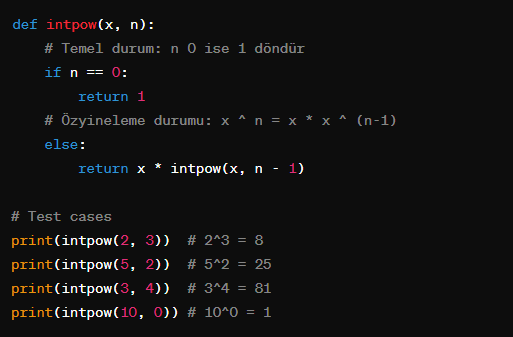
Bu özyineleme çağrı, orijinal problemin daha küçük bir versiyonunu ele almalı ve fonksiyonun her adımda problemi basitleştirerek temel duruma doğru ilerlemesini sağlamalıdır. Yani, fonksiyonun kendini tekrar çağırırken, problemin boyutunu küçültmeli ve bir sonraki adımda daha küçük bir alt sorunla ilgilenmelidir. Bu, özyineleme işleminin sonlu ve işlevsel bir şekilde devam etmesini sağlar.

13-Python'da type fonksiyonu ne döndürür? Eğer bir programda type fonksiyonunu çağırırsanız, Python'un hangi özelliğini kullanmış olursunuz?

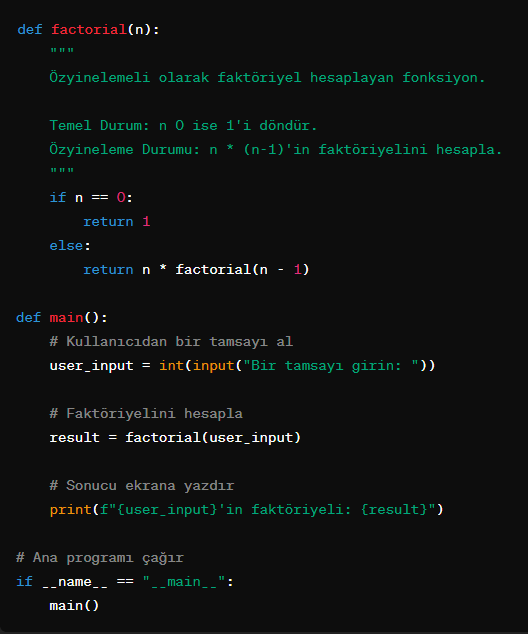
Python'da **type()** fonksiyonu, bir nesnenin türünü belirlemek için kullanılır. Bu fonksiyonu çağırdığınızda, o nesnenin türü elde edilir.

Bu özellik, Python'un dinamik türleme yeteneğini temsil eder. Dinamik türleme, değişken türlerinin çalışma zamanında belirlenmesini sağlar. Python gibi dinamik türleme dillerinde, değişken türleri açıkça belirtilmeden kullanılabilir, bu da kodun daha esnek olmasına olanak tanır.

1-Bir sayı (x) ve bir tam sayı (n) alacak şekilde intpow adlı özyinelemeli bir fonksiyon yazın ve x ^ n hesaplayın. Bu fonksiyonu tamamen başarılı sayılabilmek için özyinelemeli olarak yazmalısınız. Fonksiyonunuzun doğru çalıştığını test etmek için birkaç test örneği içeren bir program içine ekleyin.



2-Bir tamsayının faktöriyelini hesaplamak için özyinelemeli bir fonksiyon yazın. 0'ın faktöriyeli 1'dir. Herhangi bir pozitif tamsayı n'nin faktöriyeli, n'nin (n-1)'in faktöriyeli ile çarpılmış halidir. Kullanıcıdan bir tamsayı girmesini isteyen ve girilen tamsayının faktöriyelini yazdıran bir program yazın. Programınızın bir ana fonksiyonu olmasına dikkat edin. Özyinelemeli fonksiyonunuzdaki temel durumu ve özyineleme durumunu yorum satırları ile belirtin.



Bu program, kullanıcıdan bir tamsayı alır, ardından faktöriyelini hesaplamak için özyinelemeli bir fonksiyon kullanır. Temel durumu, n'nin 0 olması durumunda 1'i döndürmektir. Özyineleme durumu ise n \* (n-1)'in faktöriyelini hesaplamaktır. Programın ana fonksiyonu, kullanıcının bir tamsayı girmesini ve faktöriyelini hesaplamasını sağlar.