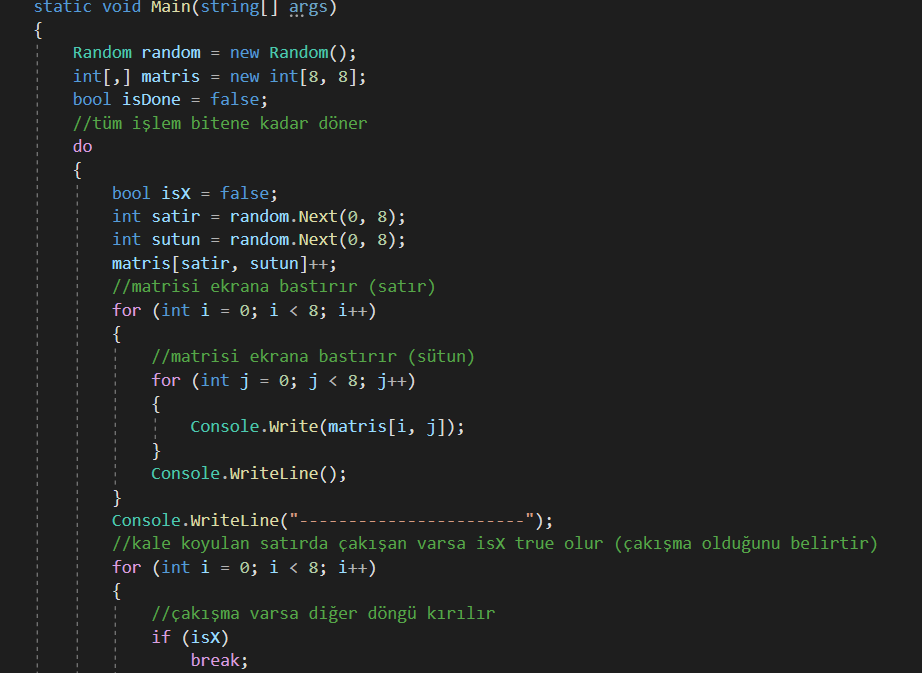
RAPOR – Soru 1

Bu programda 8 tane kaleyi bir satranç tahtasının üzerine doğru bir şekilde yerleştirene kadar gerçekleşen aşamaları ekrana yazdırmam ve satranç tahtası üzerindeki rastgele sayı üreten fonksiyon kullanarak belirlemem gerekiyor.



Öncelikle Random sınıfından random isimli bir nesne oluşturdum. Programın bazı yerlerinde random sayı bastırmak için kullanacağım Next() fonksiyonunu kullanabilmem için bu gerekliydi. Daha sonrasında int tipinde iki boyutlu matris isimli bir dizi oluşturdum. Bu dizinin her bir boyutunun toplamda 8 tane elemanı olacaktı. Satranç tahtası üzerine belli koşular sağlandığı sürece kaleler yerleştirilecekti. Bu sebeple bu koşulları kontrol etmek ve en sonunda işlem başarılı bir şekilde gerçekleştirildiğinde sona erdirmek için bir döngüye ihtiyacım vardı. For döngüsünü kullanamazdım çünkü döngünün ne kadar döneceğini bilmiyordum. Bu sebeple while döngüsü kullanmaya karar verdim fakat şöyle bir gerçek vardı ki bu döngünün öncelikle koşulları sağlasa da sağlamasa tahtaya 1 kere kale yerleştirmesi gerekiyordu. Zaten ilk tahta yerleşmediği sürece kontrol etmem gereken durumlar ortaya çıkmayacaktı. Ben de tüm programımı bir do while döngüsü içerisine yazdım. Böylelikle herhangi bir koşul gözetmeksizin tahtaya bir kere kale yerleştirilecekti. Döngünün tekrar dönüp dönmeyeceğine karar verilirken kullanılacak bool tipinde isDone isimli bir değişken oluşturdum ve varsayılan değer olarak false atadım. isDone değişkeninin true haline gelmesi demek istenilen her şey gerçekleştirildi demekti. Bu sebeple istenilen her şey gerçekleştikten sonra programdan çıkabilmek için while koşulunun içerisine (!isDone) ifadesini yazdım.

isDone false olduğu sürece while koşulunun içerisindeki ifade true olacak ve program çalışmaya devam edecekti.

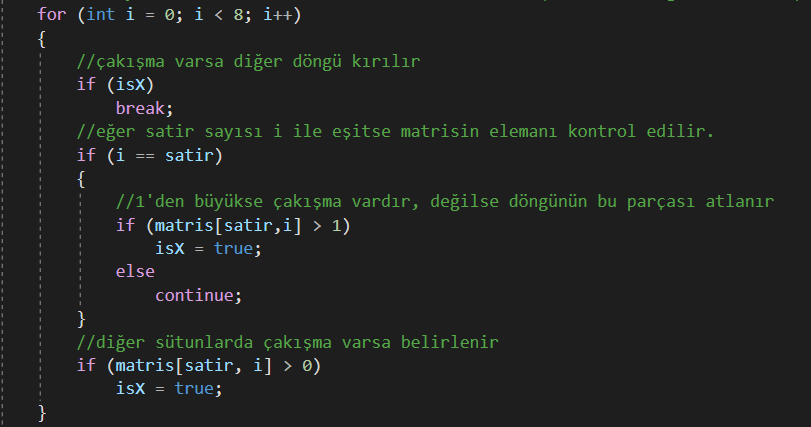
Programın amacı bir satranç tahtası üzerine 8 tane kaleyi birbirlerini yemeyecek şekilde yerleştirmektir yani aynı satır ve sütunda hiçbir şekilde ikinci bir kale bulunmamalıdır. Bundan dolayı do while döngüsünün içerisinde bool tipinde isX isimli bir değişken oluşturdum ve ilk değer olarak false atadım. isX false olduğu sürece bunun anlamı şu ana kadar herhangi bir çakışmanın olmadığıdır. Son olarak programla ilgili bahsedebileceğim şey; programda kalelerin yerleştiği yerlerin 1, diğer yerlerin ise 0 ile temsil edildiğidir. Özellikle bu iki değeri seçtim çünkü programın içerisinde kale yerleştirmek içim ++ operatörünü kullandım. Kalenin yerleşmediği yerde 0 olan değeri oraya kale yerleştirmek istediğim zaman bir arttırdığımda 1 değerini alacağı için bu değerleri tercih ettim.

Kale yerleştireceğim konumlar için daha öncesinde rastgele sayı üreten fonksiyon kullanacağımı söylemiştim. Bunu programın başında oluşturduğum random isimli nesnenin Next fonksiyonunu yaptım. Next fonksiyonunun döndüreceği değerin aralığını satranç tahtasına göre ayarladım. Şöyle anlatmam gerekirse benim en başta oluşturduğum matris isimli dizi iki boyutlu bir diziydi ve her bir boyutunda 8 eleman vardı. Bu bir tesadüf değildi. Bu diziyi 8x8 lik bir matris yani başka bir deyişle satranç tahtasını oluşturmak için kullandım. Haliyle satır ve sütun olarak ayrı ayrı düşünülürse her bir boyutta kaleyi yerleştirmek için toplam 8 alan vardı. Fakat bunu indis olarak düşünürsek 0. İndisten başlayıp 7. İndise kadar toplam 8 alan vardı . Yani ben kaleyi sekizinci alana yerleştirmek istersem bunu 7. indise yerleştirmiş olacaktım. Yani Next() fonksiyonunun bu programda üretebileceği en küçük sayı 0 en büyük sayı 7 olmalıydı. Bu sebeple ikisi de int tipinde olmak üzere satır ve sütun olmak üzere iki değişken oluşturdum ve bu değişkenlere değer olarak Next(0,8) fonksiyonun döndürdüğü değeri girdim. Yani satranç tahtasındaki bir kalenin yerleştiği yerin satır ve sütun indisi Next() fonksiyonu tarafından rastgele olarak belirlenecekti. Daha sonrasında Next() fonksiyonlarının belirlemiş olduğu koordinatlardaki matrisin elemanını bir arttırdım. Yani oraya bir kale yerleştirmiş oldum.

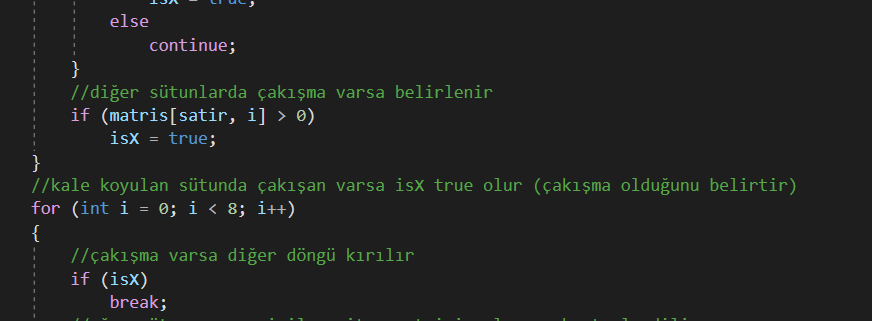
Ödevde benden 8 kaleyi doğru bir şekilde yerleştirene kadar gerçekleşen tüm adımları ekrana yazdırmam isteniyordu. Ben de bir kaleyi yerleştirdikten sonra oluşan satranç tahtasını , yani matrisi , ekrana yazdırmak için iç içe for döngüsü oluşturdum. Her bir döngünün sekiz kez dönmesini sağladım. Dıştaki döngüdeki indisin her bir değeri için içteki döngü baştan sona doğru dönecekti. Yani dıştaki döngü birinci satırı temsil ederken içteki döngü tek tek birinci satırdaki değerleri ekrana yazacaktı. Daha sonrasında döngünün ikinci satırı yazmaya başlaması için içteki döngünün bittiği yere bir Console.WriteLine() metodu yazdım. Böylelikle her bir satır bittikten sonra imleç bir sonraki satırı yazmak için alt satıra geçecekti. Dıştaki for döngüsünün bittiği yere, while döngüsü devam ettikçe yeni matrisler çizileceğinden ve matrislerin karışmaması gerektiğinden araya bir çizgi oluşturması amacıyla yeni bir Console.WriteLine() metodu yazdım.

Esasında şimdiye kadar olan bütün adımlar en azından bir kere kesinlikle yapılması gereken adımlardı. Bu andan sonraki bütün adımlarda ve döngünün sonraki turlarında her seferinde bir çatışma olup olmadığını kontrol etmem gerekiyordu. Bu, kalelerin birbirini yemeyecek şekilde yerleştirilmesi için oldukça gerekli bir adımdı. Kalelerin doğru şekilde yerleşip yerleşmediğini kontrol etmek için iki tane for döngüsü oluşturdum. Bu döngülerden ilki kalenin yerleşeceği satırı, diğeri ise sütunu kontrol edecekti.

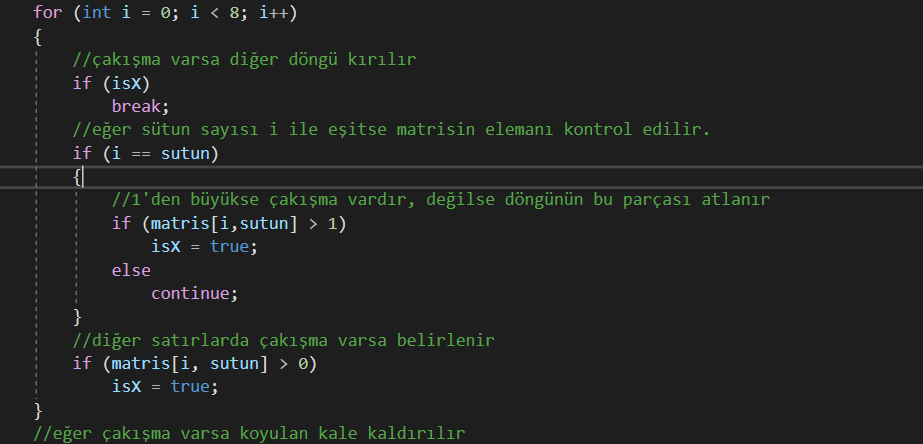
Satırı kontrol edecek olan for döngüsünün 8 kere dönmesini sağladım. Bu döngü her bir dönüşünde her bir satırı kontrol edecekti. Öncelikle bir if koşulunun içerisinde isX ifadesini kontrol ettim. Eğer isX ifadesi doğruysa if bloğunun içerisine girecek ve break komutuyla döngüyü kıracaktı. İsX ifadesinin doğru olması ise bir yerde çakışma olduğunu gösteriyordu. Tek bir satırda çakışma olması o kalenin yanlış yere yerleştirildiği konusunda yeterli bir bilgi olduğu için diğer satırlar için de döngünün dönmesini istemediğimden bu break komutunu koydum.



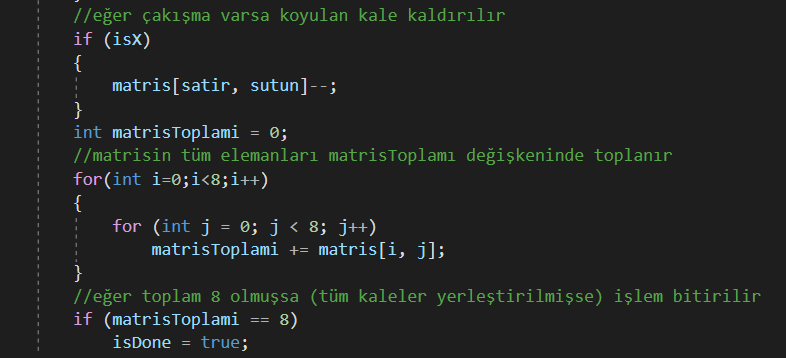
Kodu birkaç defa çalıştırdığımda fark ettim ki bazen program kale koyduğum bir yere tekrar tekrar kale yerleştiriyor. Örneğin daha önce kale yerleştirdiğim için değeri 1 olan bir yere tekrar kale yerleştirip oranın değerini 2 yapabiliyor. Böyle bir durumla karşılaşmak istemediğim için for döngüsünün içerisinde bir if koşulu oluşturdum ve bu koşulla for döngüsündeki i değerinin satır değişkenine eşit olup olmadığını sorguladım. Eğer i satır değerine eşitse yani daha önce kale yerleştirilmiş bir yere geldiysem öncelikle matrisin buradaki elemanını değerinin birden büyük olup olmadığına baktım. Bu elemanın değerinin 1’den büyük olması demek buraya birden fazla kez kale yerleştirilmiş demektir ki bu benim programımda istemediğim bir durum olurdu. Bu sebeple eğer if koşulunun içerisindeki bu ifade doğruysa çalıştırılacak olan if bloğunun içerisinde isX değişkeninin değerini true yaptım. Eğer böyle bir durum yoksa yani oraya sadece bir tane kale yerleştirilmişse çalıştırılacak olan else bloğunda ise continue komutuyla döngünün bir sonraki turuna başlamasını sağladım.



Buraya kadar, satırları kontrol ettiğim for döngüsünün içerisinde while döngüsünün o anki turunda kale yerleştirilmiş olan satırı kontrol ettim. Bu sebeple diğer satırları kontrol etmek için yeni bir if bloğu oluşturdum. Kale yerleştirilmiş bir yere yeniden kale yerleştirdiğimde değerin birden büyük olacağını biliyorduk. Fakat ben bir satırın herhangi bir sütununa bir kere kale yerleştirdikten sonra bir daha o satırdaki başka herhangi bir sütuna kale yerleştiremeyeceğim için, çünkü bu kalelerin birbirini yediği bir durum olurdu, diğer bütün durumlarda o an kontrol ettiğim sütunun sıfırdan büyük olmaması gerekiyordu. Oluşturduğum if’in ifadesi olarak bu durumu kullandım. Eğer sütunlardan herhangi biri sıfırdan büyük olursa bu çakışma olduğu anlamına gelirdi. Böyle bir durumda if’in ifadesi true haline gelecek program if bloğunun içerisine girecek ve isX değişkeni true hale gelecekti. Eğer ifade doğru değilse if bloğu atlanacak ve else bloğu çalışacak yani for döngüsünün bir sonraki adımı başlayacaktı.



Bu ilk for döngüsü bittiğinde yerleştirdiğim kaleyle beraber oluşan yeni matrisin satırlarını kontrol etmiş oldum. Bu kontrolün aynısını daha sonra ikinci bir for döngüsüyle sütunlar içinde yaptım. Orada da tıpkı ilk for döngüsünde olduğu gibi aynı adımlar izlenerek kontrol sağlanmış oldu.



Program her iki döngüden de çıktıktan sonra sıra eğer yeni koyduğum kaleyle birlikte oluşan herhangi bir çakışma varsa bu kaleyi yerleştirdiğim yerden kaldırmaya geldi.

Döngülerin içerisinde herhangi bir çakışma durumunda true hale gelecek şekilde ayarladığım isX değişenini bir if koşulunun ifadesine yerleştirdim. Eğer isX true ise yani çakışma varsa program if bloğunun içerisine girecek ve matrisin bir arttırılarak kale yerleştirilen elemanı bir eksiltilerek kale oradan kaldırılmış olacaktı. Herhangi bir çakışmanın olmadığı durumlardaysa if bloğu atlanarak programa devam edilecekti.

Program bu şekilde tüm doğru yerleri bulana kadar kaleleri yerleştirip kaldırmaya devam edecekti, ama bu program nasıl bitecekti?

Program 8 kale de doğru bir şekilde yerleştiğinde sona ermeliydi. 8 kalenin doğru bir şekilde yerleşmesi demek, matris üzerinde birbiriyle çakışmayacak şekilde yerleştirilmiş 8 tane 1 ve geriye kalan 0’lar demekti. Yani kalelerin doğru yerlere yerleştiğinden emin olduktan sonra matris elemanlarının toplamı eğer 8 olmuşsa bu benim programımın bitmesi gerektiği anlamına gelirdi.

Bu sebeple while döngüsünün en sonunda yeni bir iç içe for döngüsü oluşturdum. Matrisin her bir elemanı için dönecek bu iç içe döngünün içerisinde matrisToplami isimli bir değişken oluşturdum ve her bir matrisin değerini bu değişkenle toplayarak yine bu değişkenin içerisine yazdım.

Bu for döngüsü matrisin tüm elemanları için döndükten sonra bir if koşulu oluşturdum ve burada matrisToplamı değişkeninin değerinin 8’e eşit olup olmadığını sorguladım. Eğer sekize eşitse, program if bloğunun içerisine girecek ve programın en başında tanımladığım ve tüm işlemler bitene kadar programın dönmesini sağlayacak olan isDone değişkeninin değeri true yapılacaktı. Eğer isDone true olursa while koşulunun içerisindeki ifade false olacağı için döngü oracıkta bitecek ve program sonlanacaktı. Böylelikle programımın tüm kaleleleri doğru bir şekilde yerleştirene kadar döneceğinden emin oldum. Raporu burada sonlandırıyorum.