

[50 Puan] Alışveriş İndirimi

Bir mağazada müşterilerin özellikleri ve tutumlarına göre indirim uygulanmaktadır. Ürünler üzerinde indirimler aşağıdaki durumlara göre uygulanacaktır:

- 4 pazarlık çeşidi ve bunlara göre indirim vardır: yok(0) %0, az(1) %3, orta(2) %8, çok(3) %19
- 5 müşteri tipi vardır: çocuk(0) %2, genç(1) %10, üniversite öğrencisi(2) %15, orta yaşlı(3) %0, yaşlı(4) %1

Not: Eğer müşteri üniversite öğrencisiyse aynı zamanda genç sayılmakta ve bu iki indirim birden uygulanmaktadır.

- 3 müşteri davranışı vardır: kaba(0) %-10, nötr(1) %0, kibar(2) %5

Not: Negatif yüzde, ürün fiyatında artış anlamına gelmektedir, yani kaba müşterilere bindirim (fiyat artışı) uygulanmaktadır.

- 10 ürün tipi vardır, ürün numaralandırması 0'dan başlamaktadır ve 0. ve 2. ürünlerde hiçbir indirim yapılmamaktadır. 5. ve 8. ürünlerde ise ürün fiyatı üzerinden maksimum toplam %25 indirim yapılabilmektedir.

Not: İndirimlerin direkt olarak toplanmayacağına dikkat edin. Örneğin %15 indirim ardından uygulanan %10 indirim, ürün üzerinde toplam %25 indirim uygulandığı anlamına **gelmez**.

- Mağazadaki en pahalı ürün 5000 ₺'dir.
- Kaba müşterilere uygulanacak indirimlerden sadece en büyük yüzdeli olanı (varsa) yapılmakta, daha sonra üzerine bindirim yapılmaktadır. Ayrıca, ürünün bindirimli fiyatı maksimum ürün fiyatı olan 5000 ₺'yi geçemez.

Not: Birden fazla indirim alan müşterilere indirimli fiyat üzerinden tekrar indirim yapılmaktadır. (Örneğin %19 indirim yapılsın, daha sonra %10'luk bir indirim daha varsa bu indirim, ürün fiyatından değil, %19 indirilmiş fiyat üzerinden yapılmalıdır.)

Bu özelliklere bakarak (parantez içinde belirtilen sayılar ve yanlarında yazan indirim miktarlarına göre) aşağıdaki girişleri alan ve **indirimli_fiyat** çıkışı veren devreyi Verilogda Davranışsal Modelleme kullanarak tasarlayınız. Oluşturacağınız modüle **indirim** (oluşacak dosya **indirim.v**) ismini verin.

- **urun_fiyati** girişi
- **pazarlik** girişi
- **musteri_tipi** girişi
- **musteri_davranisi** girişi
- **urun_tipi** girişi
- **indirimli_fiyat** çıkışı

Not: **urun_fiyati** girişi her zaman tam sayı olarak gelecektir, **indirimli_fiyat** çıkışı ise küsürlü olabilir ve **indirimli_fiyat** çıkışının **en anlamsız bitleri** (sabit bit sayısında olacak), 0'dan 100'e kadar olan kuruşluk kısmını göstermelidir. (Kuruşluk kısmında hep aşağı yuvarlayın, örneğin indirimli sayıda 2,8 kuruş varsa **indirimli_fiyat** çıkışının tam sayıdan sonra gelen küsürat kısmında 2 kuruş olmalıdır.)

Not: Yukarıda verilenlere göre giriş çıkış bit sayılarını minimum sayıda olacak şekilde sizin bulmanız gerekmektedir.

Örnek: **urun_fiyati**=25, **pazarlik**=2, **musteri_tipi**=2, **musteri_davranisi**=0, **urun_tipi**=7 ise **indirimli_fiyat**=23.37 olmalıdır -> Üst bitler 23 ₺'yi, alt bitler ise 37 kuruşu gösterir. (Kaba müşterilere sadece en büyük yüzdeli bir indirim uygulandığına dikkat edin, örnekten de anlaşılacağı üzere, üniversite öğrencisi olmasına rağmen ekstra gelen genç indirimi uygulanmamıştır.)

[50 Puan] Petrol İstasyonu

Bir petrol istasyonunda tadilat gerçekleşmiş ve 200 bin ₺ tutmuştur ve bu para 30 gün içinde amorti edilmek istenmektedir.

Bu petrol istasyonunda yapılabilecek 4 çeşit işlem vardır. Bunlar aşağıdaki gibi sıralanabilir.

- 00 - Yakıt almak (işlem başına ortalama 250 ₺ kar)
- 01 - Lastik basıncı kontrolü (işlem başına ortalama 0 ₺ kar)
- 10 - Araç yıkamak (işlem başına ortalama 50 ₺ kar)
- 11 - Market alışverişi yapmak (işlem başına ortalama 30 ₺ kar)

Her bir gün için; ortalama verilen araç sayısı ve o gün petrol istasyonunda en çok gerçekleşen işleme (o gün için geri kalan işlemler önemsenmiyor) ve işlem başı ortalama kara bakılarak araç sayısı ile çarpılıp günlük kar hesaplanmaktadır. Yalnız bir istisna vardır, eğer bir önceki gün en çok yapılan işlem **araç yıkama** işlemi ise, şu an bulunulan gün işlem girişi ne gelirse gelsin, en çok yapılacak işlemin **yakıt almak** olacağı biliniyor.

Bu petrol istasyonunu Verilogda Davranışsal Modelleme kullanarak tasarlamamız bekleniyor. Oluşturacağınız modüle **istasyon** (oluşacak dosya **istasyon.v**) ismini verin. Devrenin giriş ve çıkışları aşağıdaki gibidir.

- Günlük ortalama araç sayısını belirten **ARAC_SAYISI** parametre girişi (varsayılan değer 200 olsun.)
- **saat** girişi
- **reset** girişi
- 2 bitlik **islem** girişi
- Tadilat parasının geri kazanıldığını belirten 1 bitlik **amorti** çıkışı (başlangıçta 0, amorti olduğu günden itibaren 1)
- Kaçınıcı günde amorti olduğunu belirten 5 bitlik **amorti_gunu** çıkışı (eğer 30 günün sonunda amorti olmadıysa 0 olarak kalmalıdır.)

Her bir günü bir saat vuruşu olarak varsayın ve her yükselen saat darbesinde durum değişikliklerini gerçekleştirin. 30 gün sonunda harcanan para hala amorti edilemediyse devam eden günlerde de amorti çıkışı 0 olarak kalmalıdır. reset sinyali geldiğinde devre başlangıç durumundan, yani 1. günden başlamalıdır.

Ödev Teslimi (Son Teslim Tarihi: 31.07.2023 16.30)

1-) indirim.v

2-) istasyon.v

dosyalarını **sıkıştırmadan** <https://uzak.etu.edu.tr>'ye yükleyin.