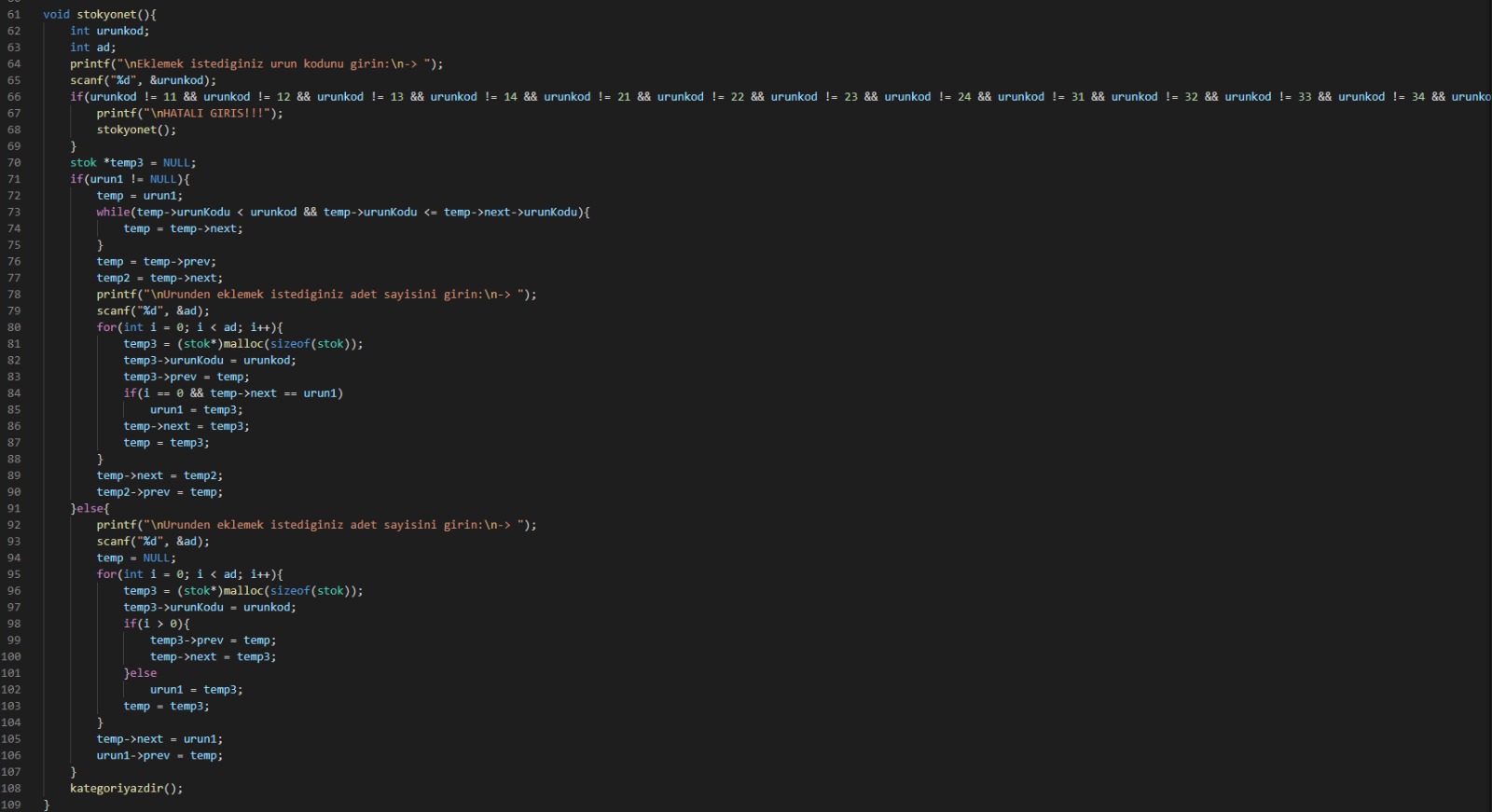
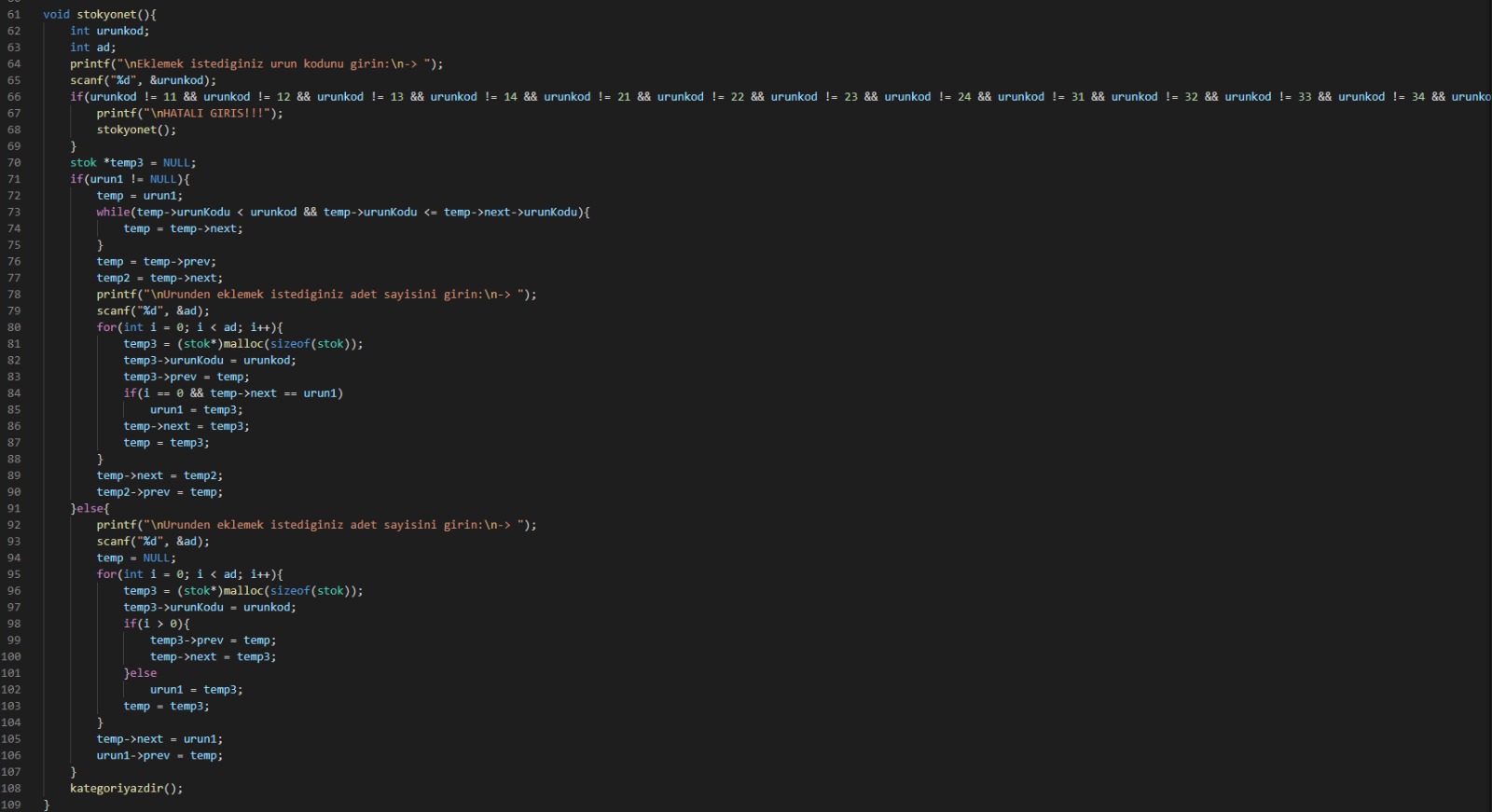
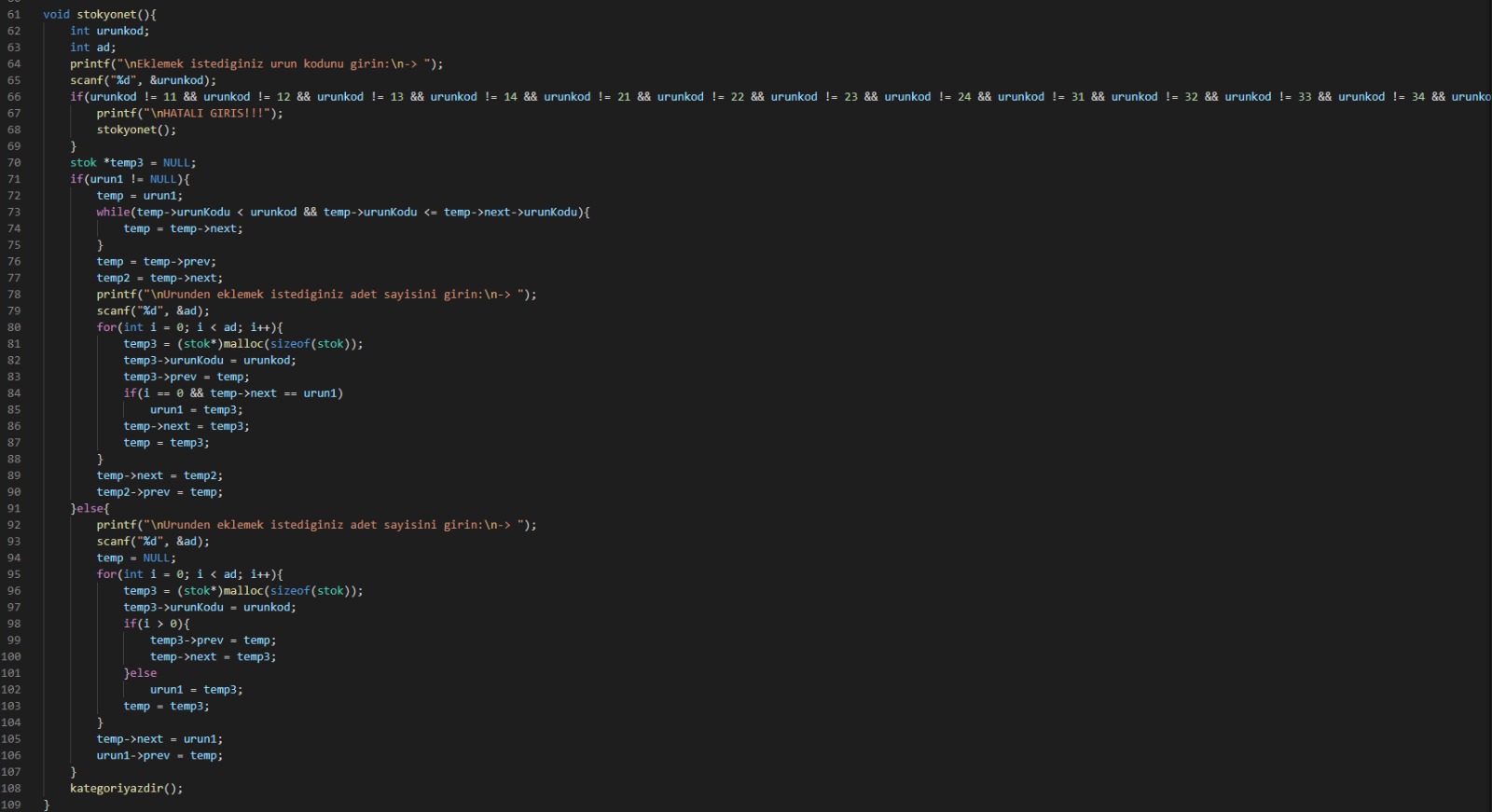
stokyonet

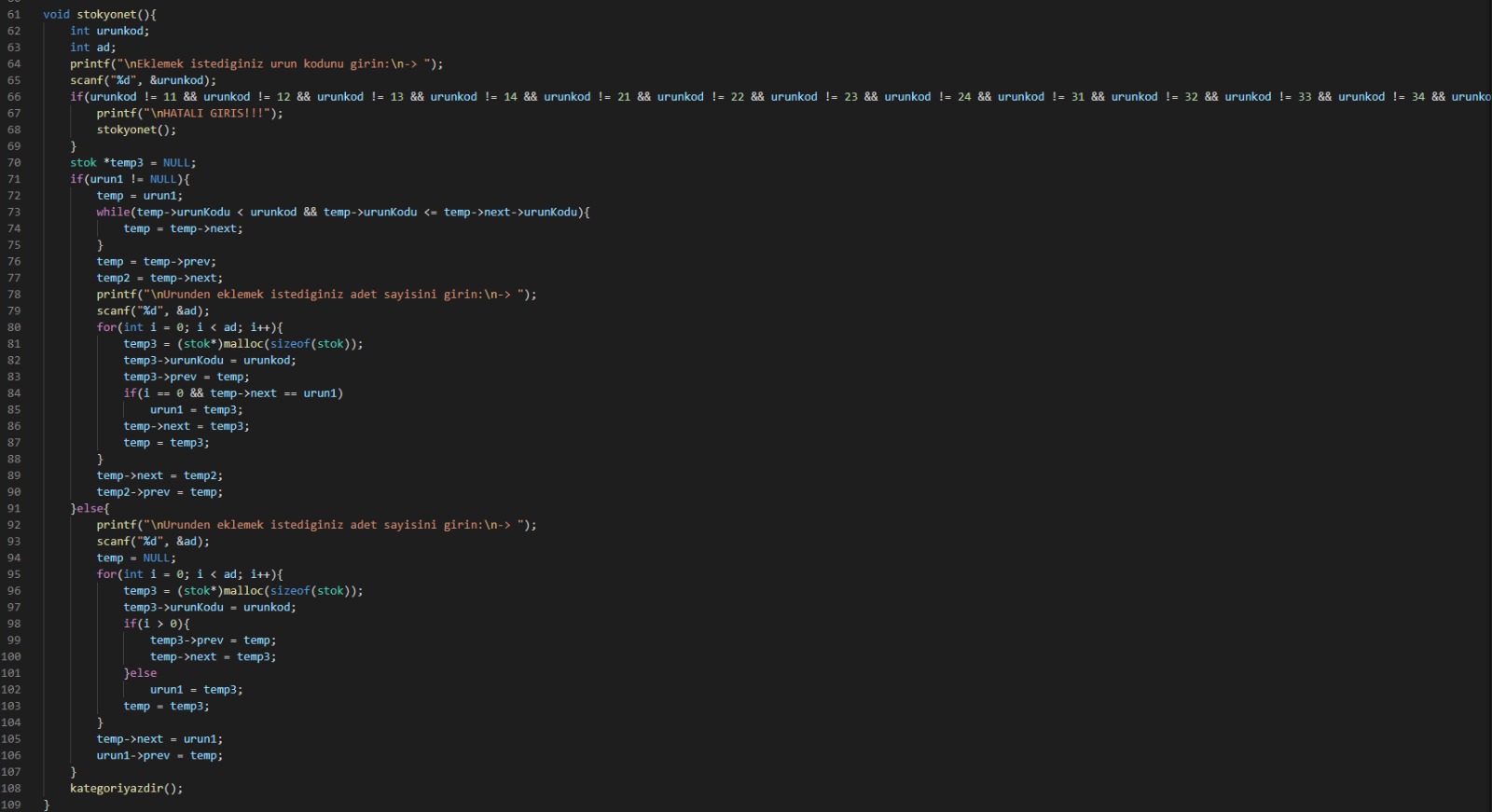
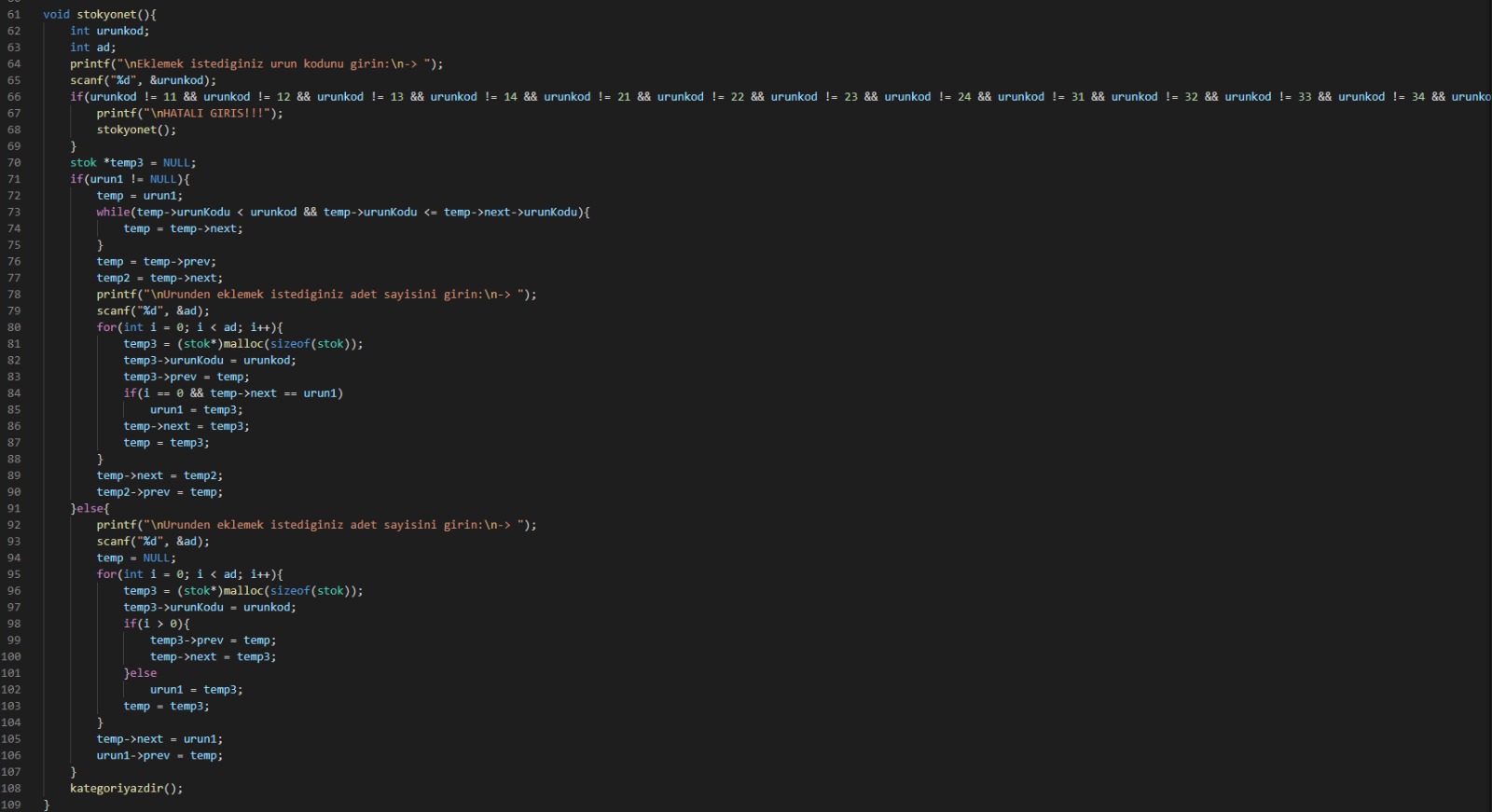
Kullanıcıdan ürün kodu alınıyor.

Girilen kod hatalı ise, hata mesajı ver.

urun1 null değil ise(stok boş değil):

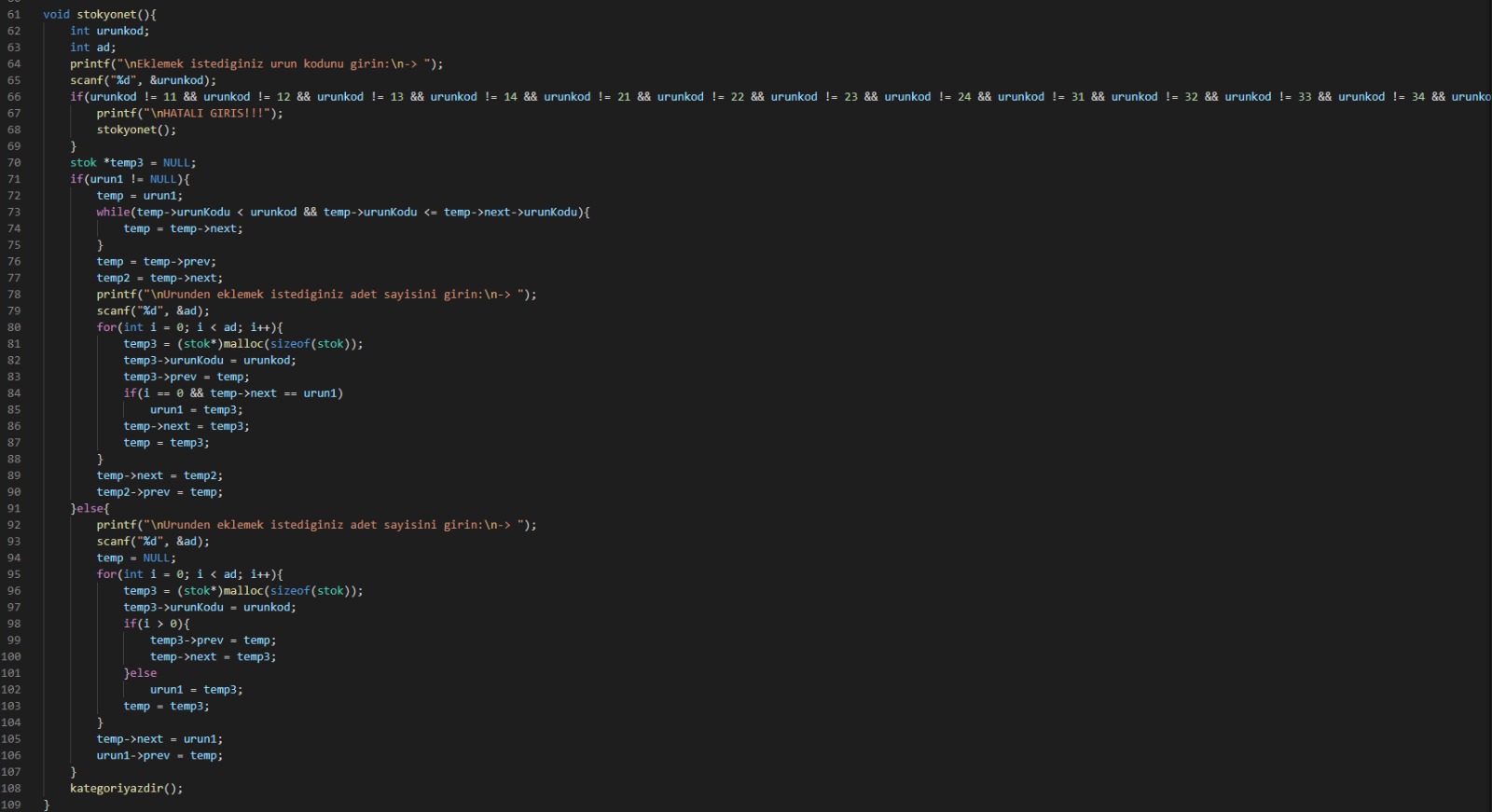
-temp’i urun1’e eşitle

-girilen ürün kodu temp’in ürün kodundan büyük olduğu sürece temp’i temp’in next’ine eşitle(ürünü koduna göre yerleştirmek için). (temp ile girilen ürün kodundan daha küçük ve temp2 ile eşit veya daha büyük ürün koduna sahip ürünü tutuyor)

-kullanıcıdan eklemek istediği adet sayısını iste.

-stoğa ürünleri ekle.

-Eğer eklenen ürünün kodu ürün1 koduna eşit veya daha küçük ise, ürün1’i eklenen ilk ürüne eşitle(eklenen ürün kodu en küçük iken stoğun sonunda kalmaması için).

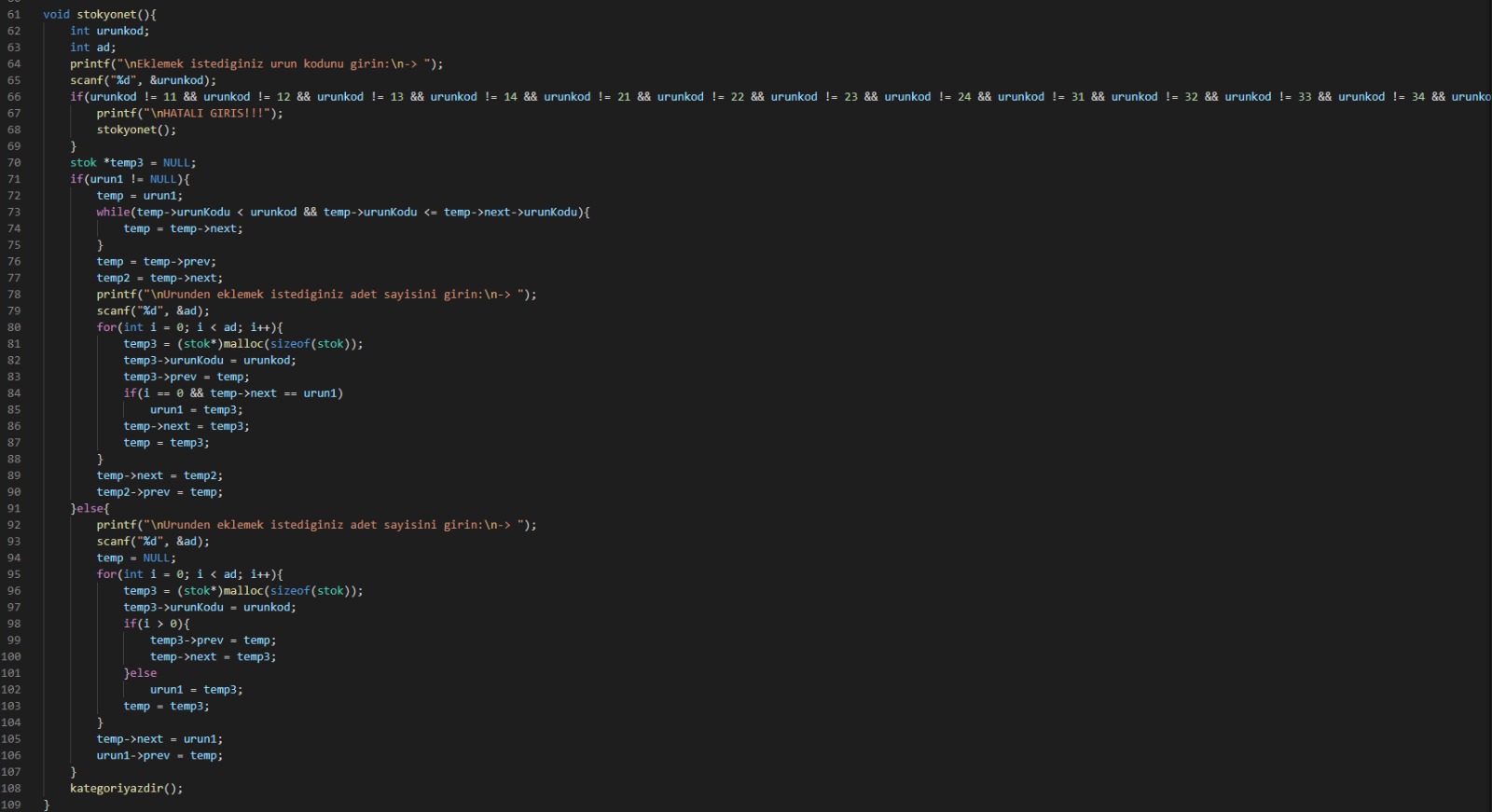
-son eklenen ürünün(temp) next’ini temp2’ye, temp2’nin prev’ini temp’e bağla.

Urun1 null ise(stok boş):

- kullanıcıdan eklemek istediği adet sayısını iste.

- stoğa ürünleri ekle. Eklenen ilk ürünü urun1’e eşitle.

- son eklenen ürünün(temp) next’ini urun1’e, urun1’in prev’ini temp’e bağla.

kategoriyazdir menüsüne dön.

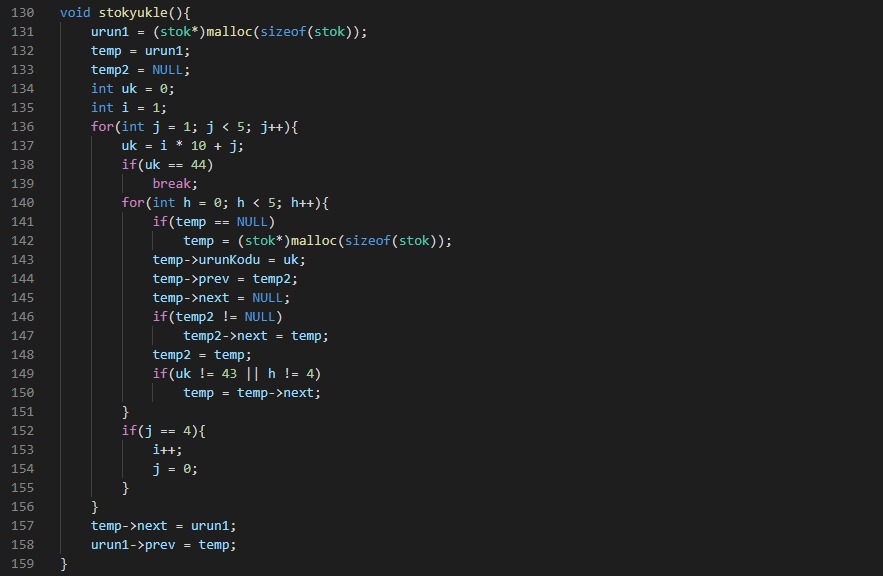
kategoriolustur

Geçici iki bağlı liste oluştur.

Kategoriler dizisinin elemanlarına uygun ürün kodu ata.

Dizinin her elemanın next’ine aynı kategorideki elemanları ekle.

4. kategori sadece 3 üründen oluştuğu için i ve j 3’e eşit olduğunda dögüden çıkılıyor.

stokyukle

ürün1’e bellekte yer açılıyor.

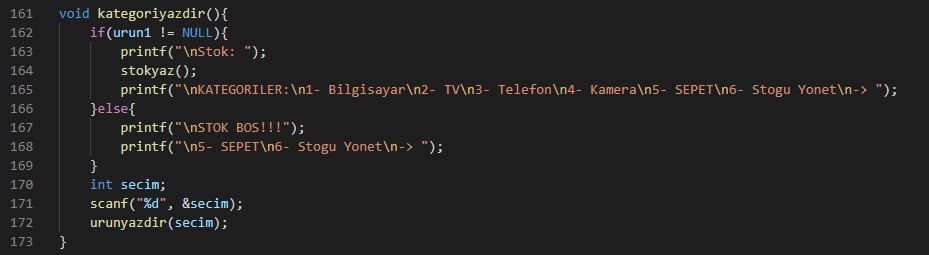
Kategoriler 4 üründen oluştuğu için j yi 1 eşitleyip j<5 koşuluyla döngü çalışıyor.

4. kategori 3 üründen oluştugu için uk 44’e eşit olunca döngünün başından çıkılyor.

Her üründen 5 tane oluşturuluyor.

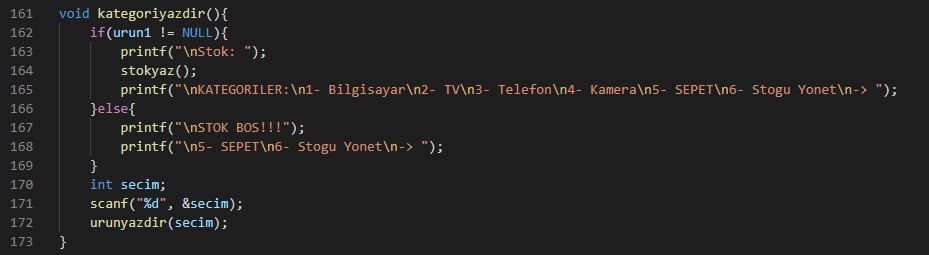
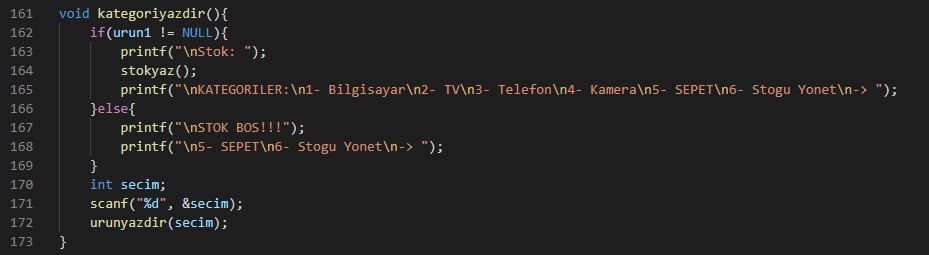
j 4’e eşit olunca, j sıfırlanıp i bir artırılıyor. Böylece bir sonraki kategoriye geçiliyor.

Son ürünü(temp) ürün1’e bağlayıp(next/prev), bağlı liste dairesel oluyor.

kategoriyazdir

Stok boş değil ise:

Stokyaz fonksiyonuyla stoktaki elemanların ürün kodlarını yazdır.

Kategoriler yazdırılıyor.

Stok boş ise: Stok boş mesajını verip, sadece sepet ve stoğu yönet yazdırılıyor.

Kullanıcıdan seçim alınıyor.

urunyazdir

Seçime göre:

1/2/3/4 için:

Kategori ismi yazdırılıyor.

kategoriler dizisindeki uygun eleman seçilip,

o kategorideki her elemanın ürün kodu alınıp,

urunk fonksiyonuyla ürünün ismi yazılıyor.

Kullanıcıdan seçim alınıyor.

Seçim 5(geri) ise kategoriyazdır fonksiyonuna geri

gönderiliyor.

1 ve 4 arasında değil ise hata verip urunyazdir

fonksiyonu tekrar başlatılıyor.(4. Kategori için 1 ve 3

arası değil ise)

Stokara fonksiyonu kategori numarası ve ürün numarası

İle çağırlıyor.

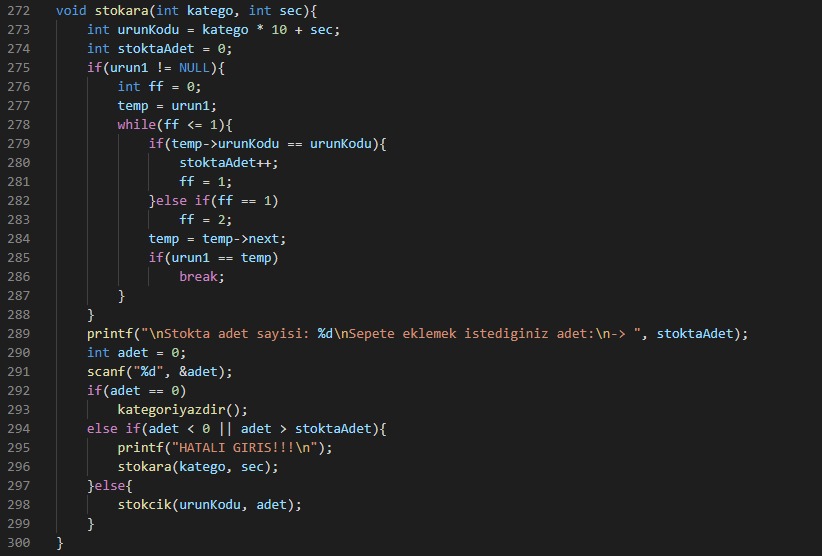
5 için: sepet fonksiyonuna gönderiliyor.

6 için: stokyonet fonksiyonuna gönderiliyor.

default(bunlardan hiç biri ise): hata mesajı verip

kategoriyazdir fonksiyonuna dön.

stokara

Stok boş değil ise:

temp’i ürün1’e eşitle ve ff 0’a eşit.

ff 1 den küçük veya eşit olduğu sürece:

temp’in ürün kodu istenilene eşit ise

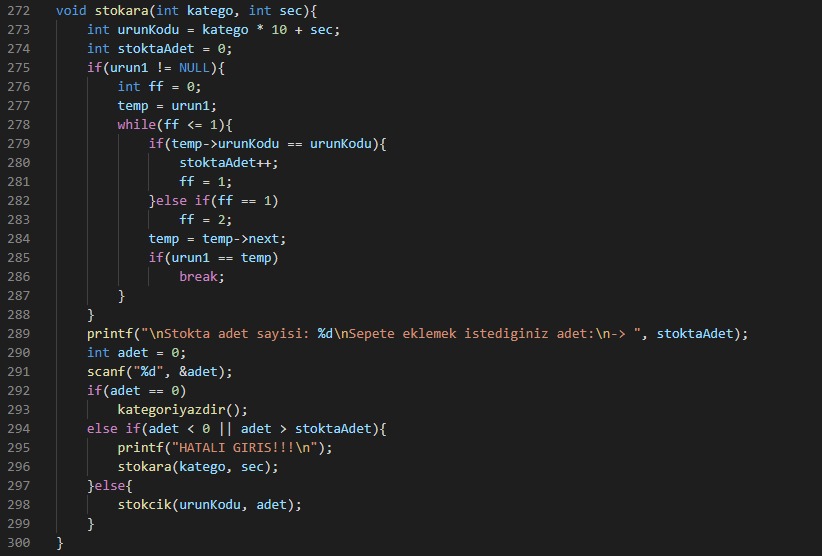
stoktaAdet’i 1 artır ve ff’i 1’e eşitle.

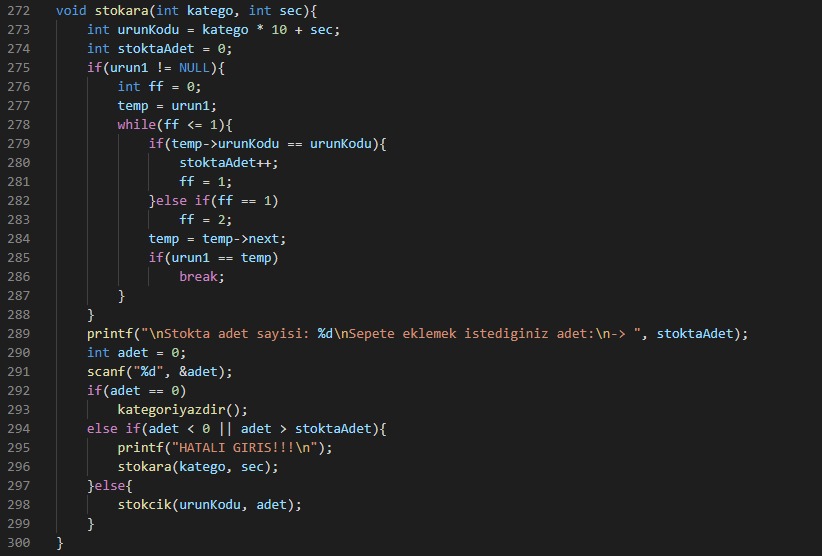
Eşit değil ise ve ff 1 ise ff’i 2’ye eşitle.(dögüyü

durdur)

temp’i temp’in next’ine eşitle.

temp urun1’e eşit ise döngüyü durdur.(tüm

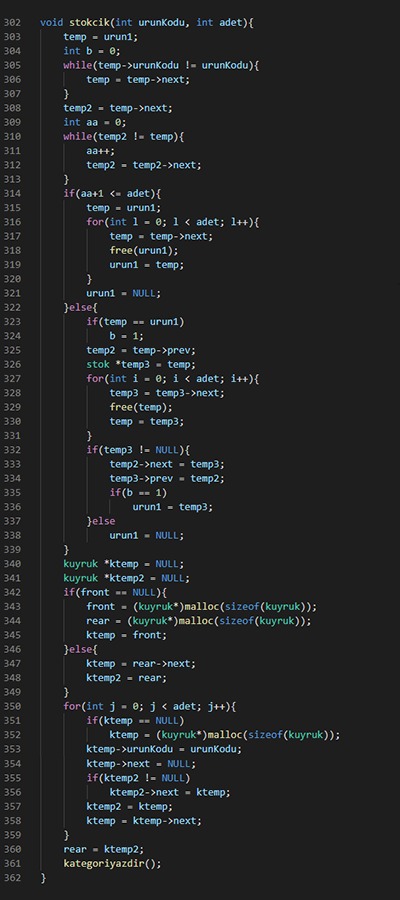
 ürünleri gözden geçirmiş oluyoruz)

Stoktaki istenilen ürünün adet sayısını yazdırp, kullanıcıdan kaç tane istediği soruluyor.

İstenilen adet 0 ise kategoriyazdir’e geri dön.

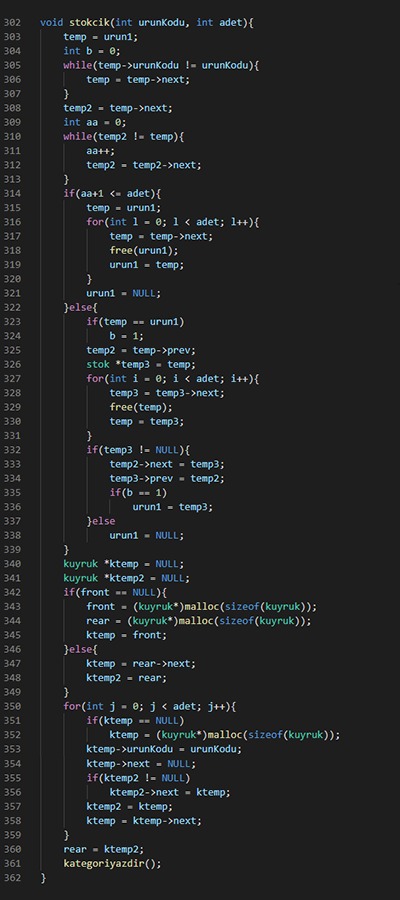
Negatif veya stoktaki adet sayısından fazla istendiyse hata mesaajı verip stokara fonksiyonu tekrar başlatılıyor.

Önceki şartlar sağlanmıyorsa(düzgün sçim yapılmış),stokcik fonksiyonu ürünkodu ve adet sayısıyla çalıştırılıyor.

stokcik

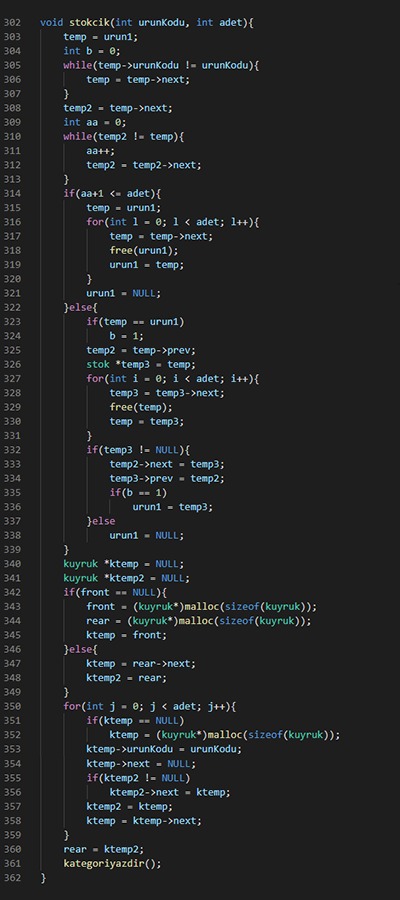
temp’i ürün1’e eşitleniyor.

temp’in ürün kodu istenilene eşit olana kadar, temp temp’in next’



Stokta kaç ürünün olduğuna bakılıyor.

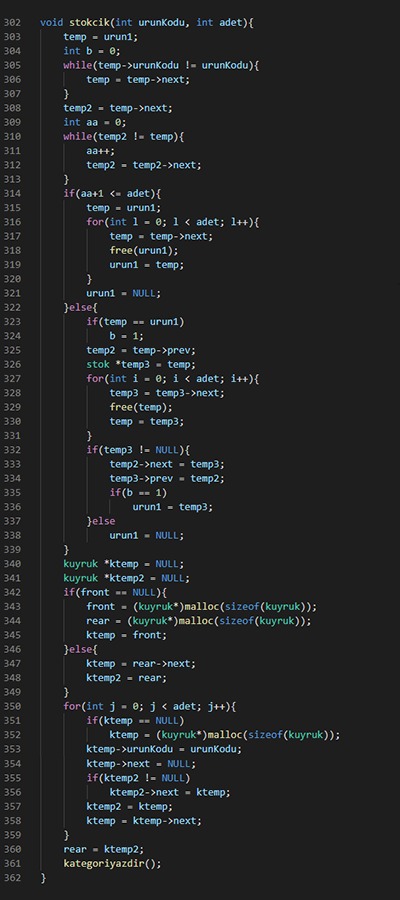
Eğer istenilen adet sayısı stoktaki adet sayısına eşit ise(stokta kalan tüm ürünler isteniliyor anlamına geliyor), stoğu boşalt ve urun1’i null’e eşitle.

temp urun1’e eşit ise(stoktaki ilk üründen isteniliyor), b’yi 1’e eşitle.

istenilen kadar stoktan çıkarılıyor.

Eğer temp3 null değil ise, temp2’yi temp3’e bağla ve b 1 ise urun1’i temp3’e eşitle.

temp3 null ise(stok boşaldı), urun1’i null’a eşitle.

kuyruk boş ise, front ve rear için bellekte yer ayır ve ktemp’i front’a eşitle.

Boş değilse, ktemp’i rear’in next’ine ve ktemp2’yi rear’e eşitle.

İstenilen adet kadar üründen kuryruğa ekle.

rear’i ktemp2’ye(kuyruğun son elemanı) eşitle.

Kategoriyazdir’a dön.

sepet

Daha önce teslim edilen ürünleri yazdır.



Sepet boş değil ise,

sepetteki ürünleri yazdır.

 -kullanıcıdan sepetti onaylanması isteniliyor.

Onaylarsa kuyrukcik fonksiyonuna

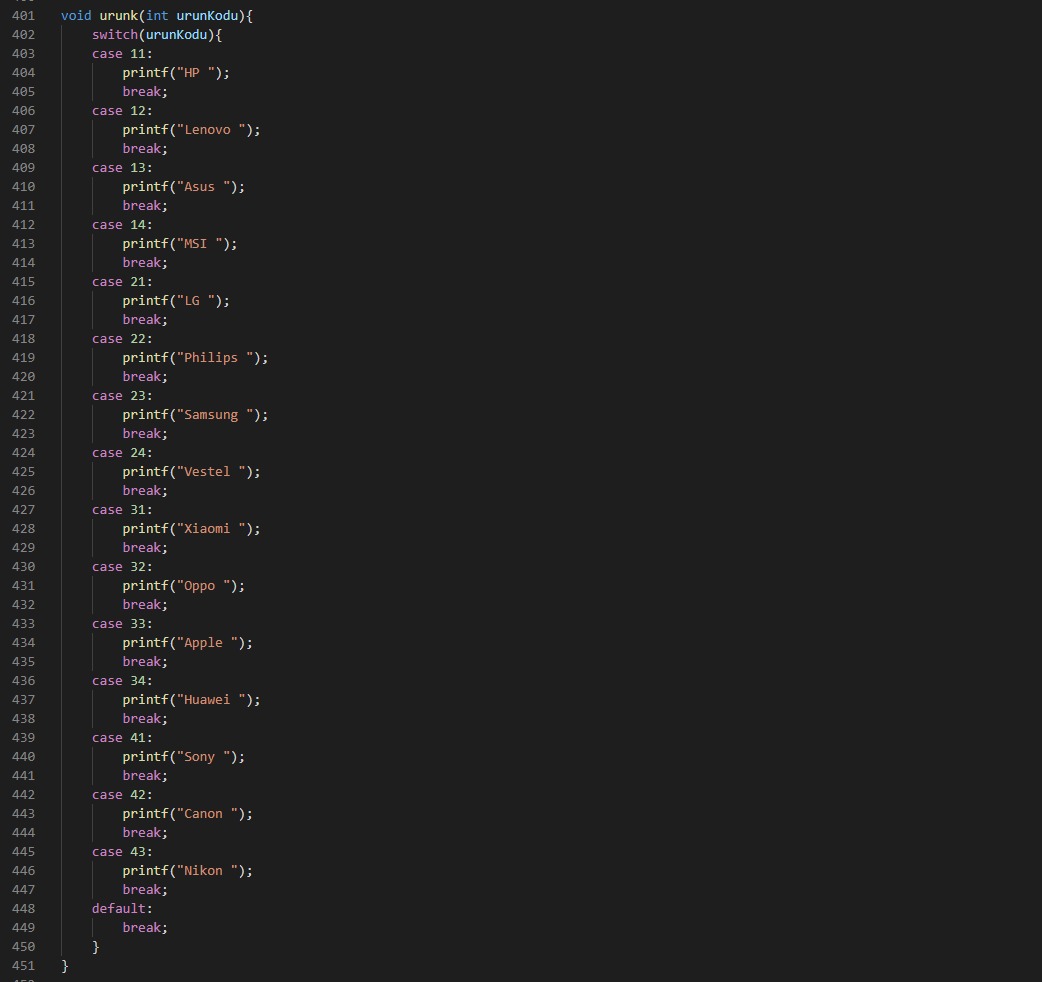
gönderiliyor.

Geri derse kategoriyazdir’a dönülüyor.

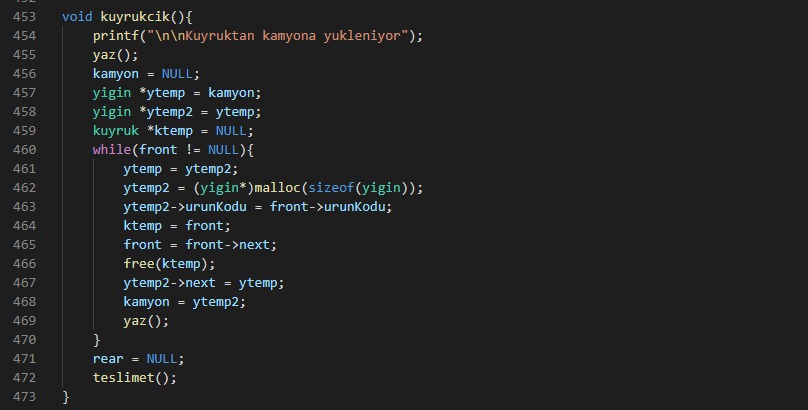
İkisi de değil ise hata mesajı verilip sepet

fonksiyonu tekrar çalıştırlıyor.

Sepet boş ise, sepet boş mesajı yazdırılıp kategoriyazdir’a dönülüyor.



Alınan urunkodunun ismini yazdırır.

kuyrukcik

Kuyruk boş olamdığı sürece,

ilk elemanını çıkarıp kamyonun tepesine ekleniyor,

yaz fonksiyonu ile kuyruk ve kamyondaki elemanlar yazdırılıyor.

yaz

Kuyruktaki elemanlar yazdırılıyor.

Kamyondaki elemanlar yazdırılıyor.

yaz2

Kamyondaki elemanlar yazdırılıyor.

Teslim edilenlerdeki elemanlar yazdırılıyor.

teslimet

Eğer user bağlı listesi(teslim edilen listesi) null ise(daha önce bir şey teslim edilmemiş), bellekte yer ayrılıyor, btemp ve btemp2’yi user’a eşitleniyor.

Eğer user null değil ise, btemp2’yi listenin son elemanına ve btemp’i onun next’ine eşitleniyor.

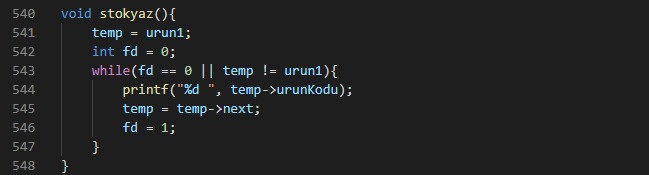
Kamyon boş olmadığı sürece,

btemp null ise bellekte yer ayrılıyor,

kamyonun tepesindeki ürünü çıkarıp teslim edilen listesinin sonuna ekleniyor,

yaz2 fonksiyonu ile kamyon ve teslim edilendeki elemanlar yazdırılıyor.

Kategoriyazdir’e dönülüyor.

stokyaz

temp urun1’e eşitleniyor.

temp urun1’e tekrar eşit olasıya kadar,

temp’in ürün kodu yazdırılıp, temp next’ine eşitleniyor.