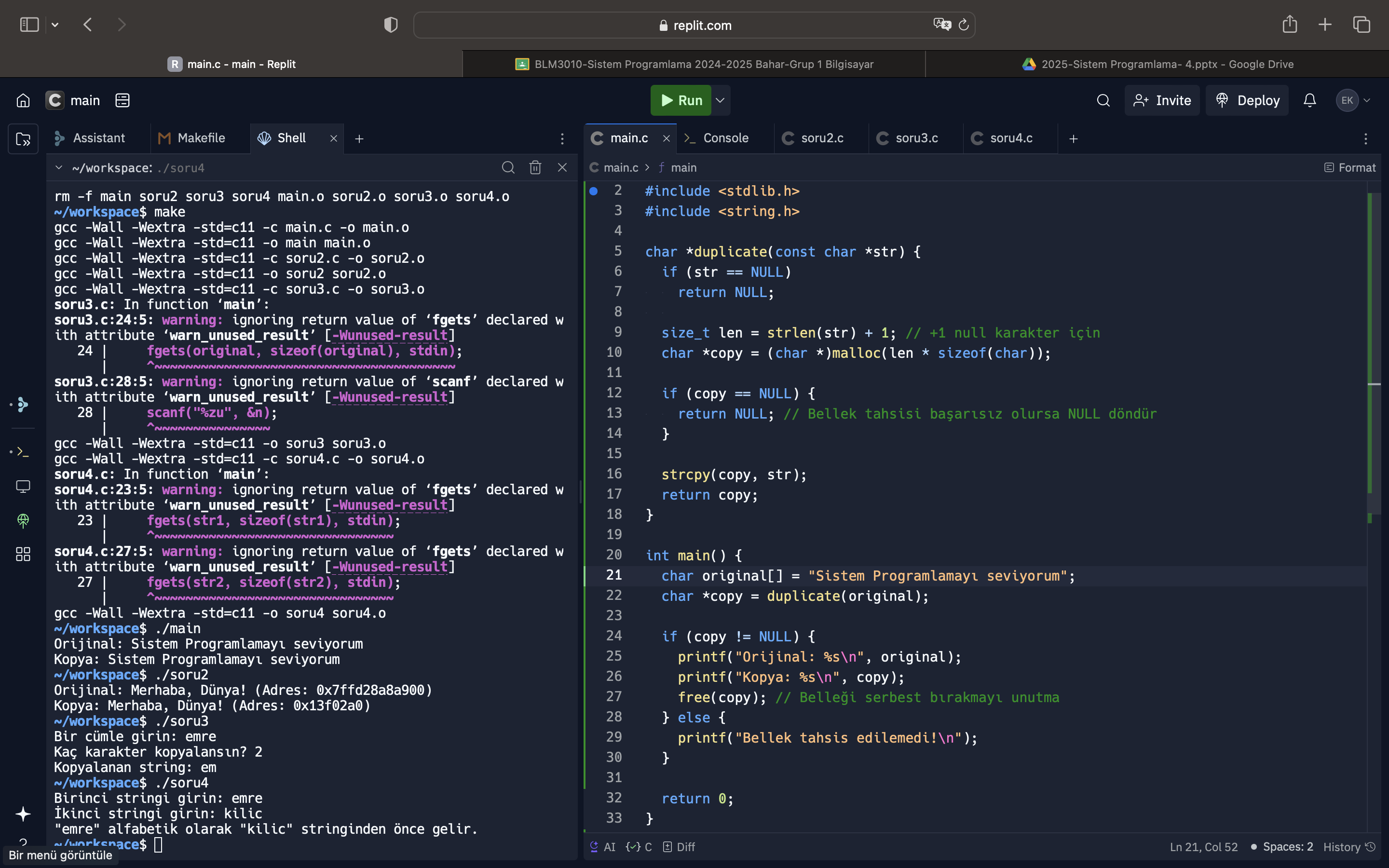
Bu Lab uygulaması için online bir platform olan Replit’i kullandım. Ödev içeriğinde sorular ve çıktıkları, kod yorumları, ve son olarak makefile bulunmaktadır.

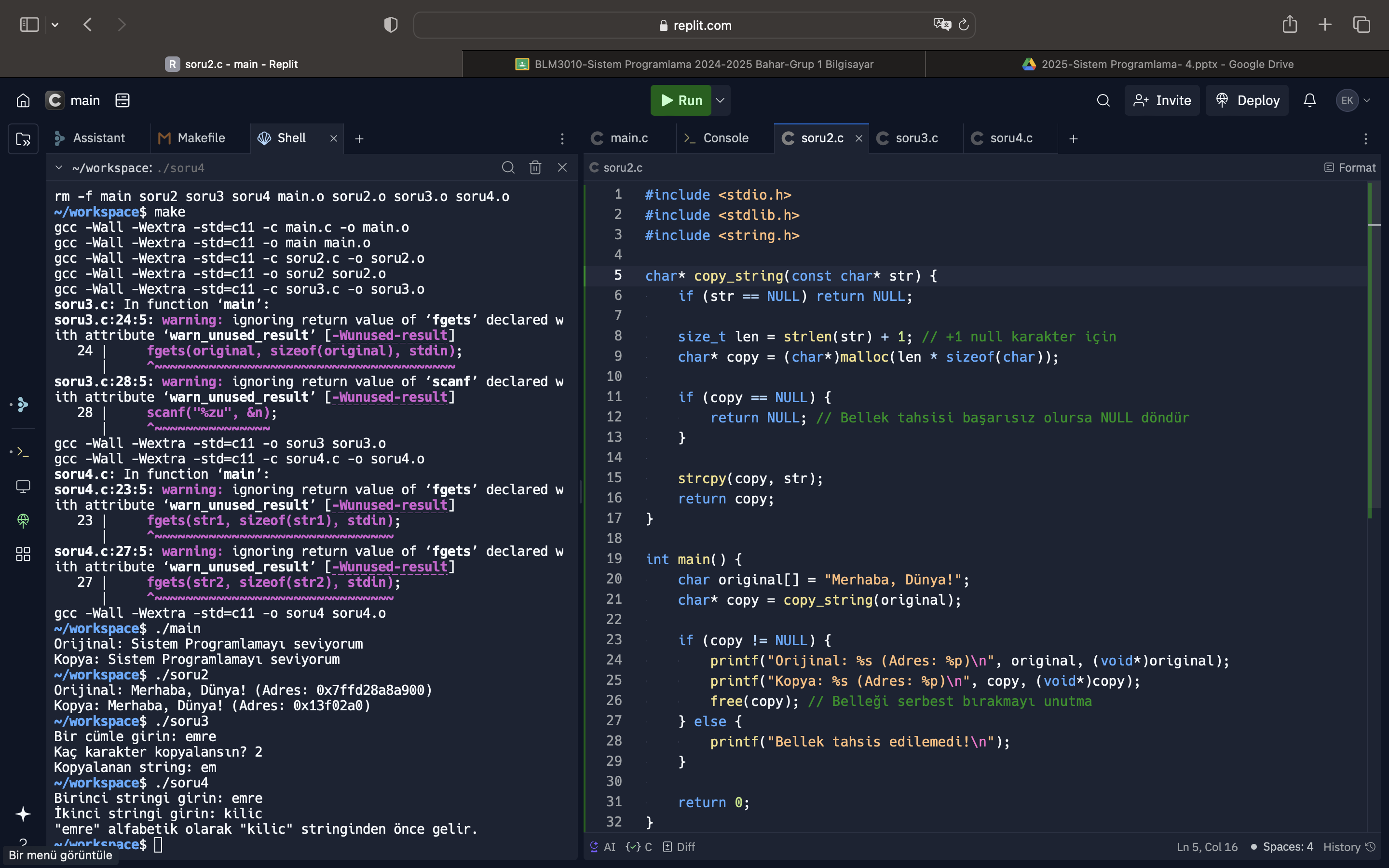
1.Soru



Bu soruda, bir stringin bellekte yeni bir kopyasını oluşturan bir fonksiyon yazmam gerekiyordu. Bunun için önce **malloc** ile yeni bir bellek alanı tahsis ettim ve **strcpy** fonksiyonunu kullanarak orijinal stringi bu alana kopyaladım. Eğer bellek tahsisi başarısız olursa NULL döndürerek hata kontrolü yaptım. Programın mantığı, stringin içeriğini değiştirmeden yeni bir kopyasını oluşturup döndürmekti.

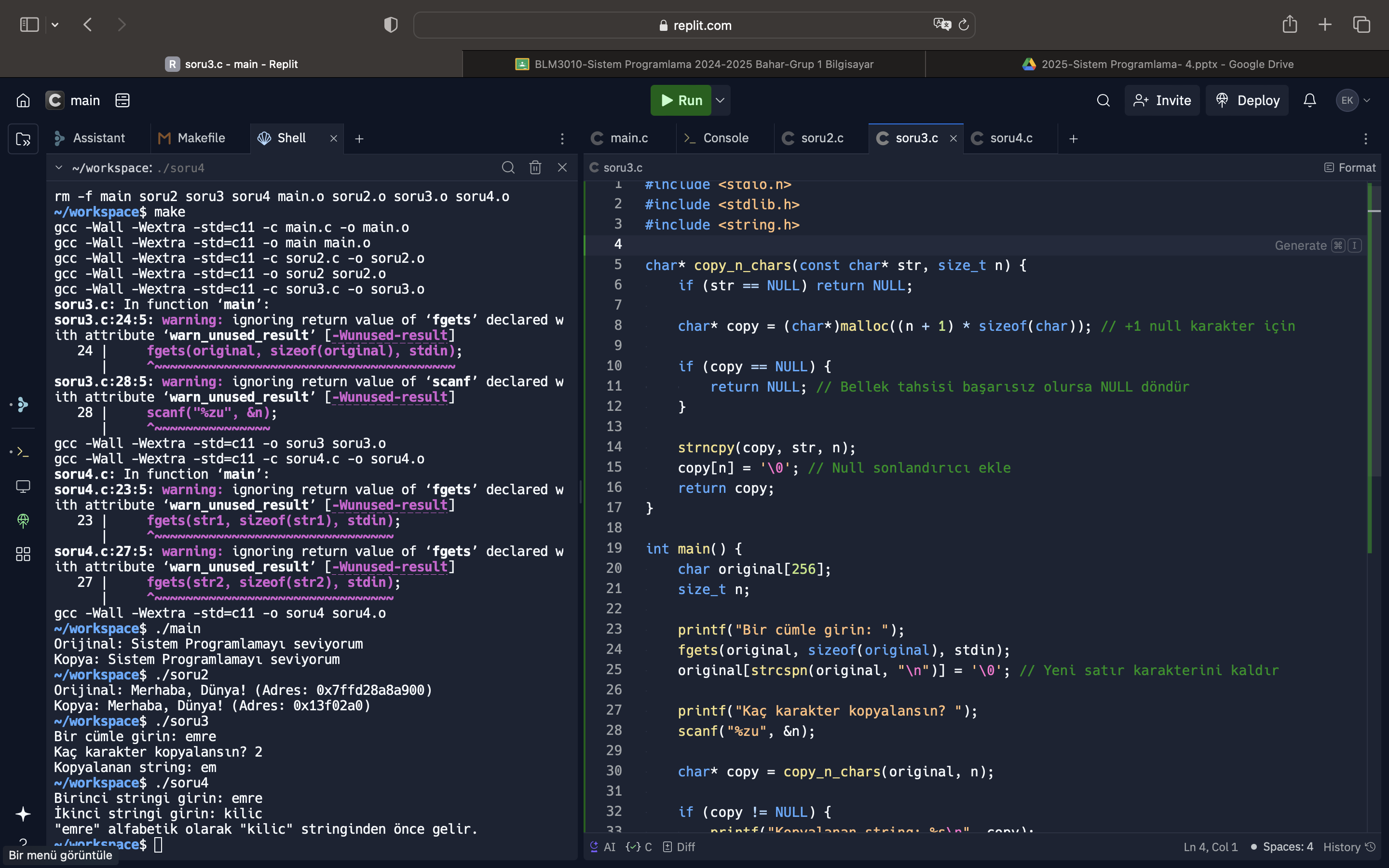
Bu çalışmayı yaparken bellek tahsisinin önemini ve **malloc** ile ayırdığımız belleği **free()** ile serbest bırakmanın gerektiğini daha iyi anladım. Eğer **free()** kullanmazsak bellek sızıntısı olabilir.

2.Soru



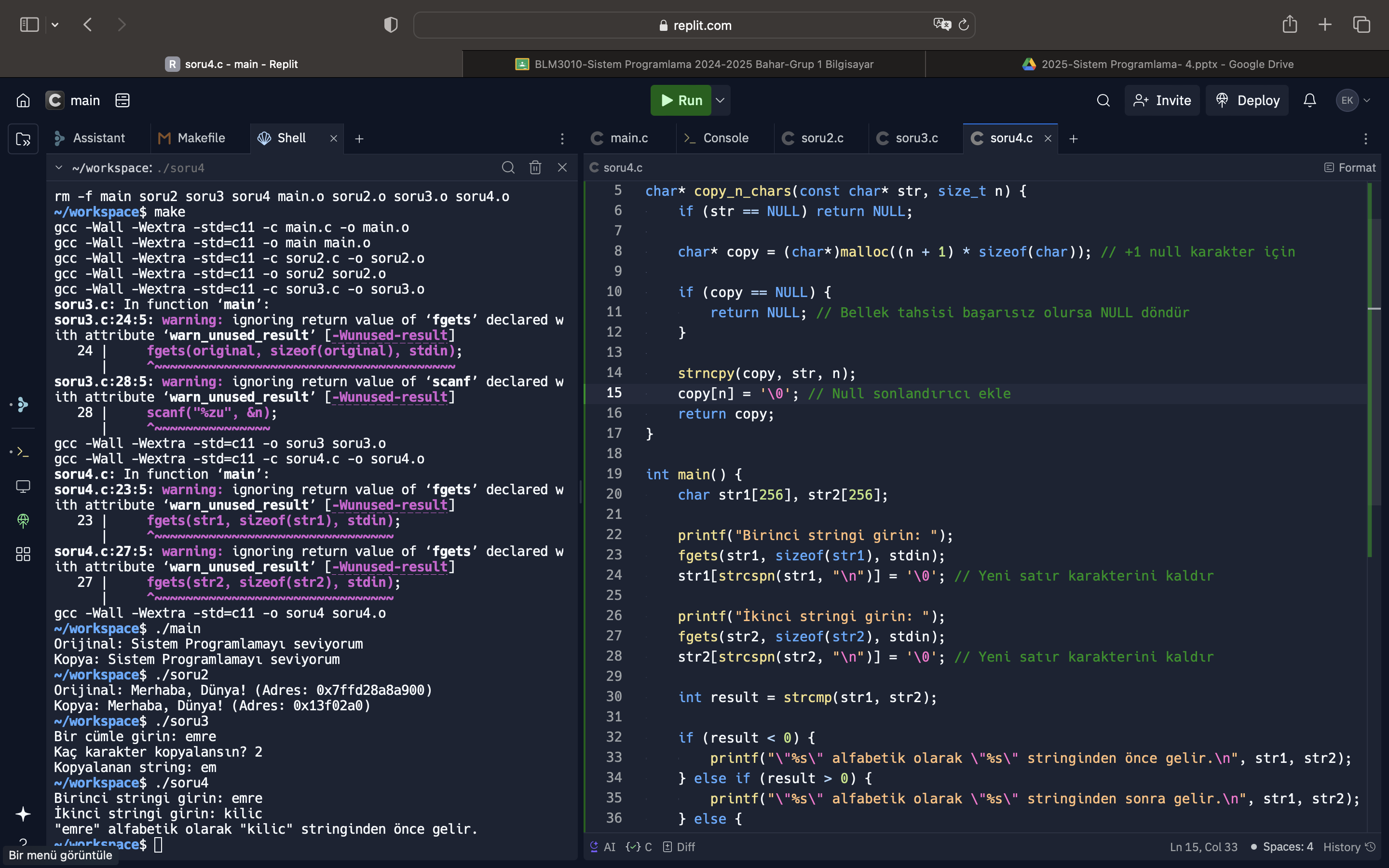
Bu soruda, kullanıcının girdiği bir cümlenin ilk N karakterini alıp başka bir stringe kopyalamam gerekiyordu. Burada **strncpy** fonksiyonunu kullanarak kopyalama işlemini gerçekleştirdim. Kullanıcının girdiği N değeri kadar karakter kopyalandı ve sonuna \0 eklenerek stringin düzgün sonlandığından emin oldum. Ayrıca, kullanıcıdan N değeri aldığım için, N değeri cümle uzunluğundan büyük olursa hata olmaması için **malloc** ile dinamik bellek yönetimi yapmanın önemli olduğunu düşündüm.

3.Soru



Bu soruda, kullanıcının girdiği iki stringi **strcmp** fonksiyonunu kullanarak karşılaştırmam gerekiyordu. Eğer **strcmp** sıfır döndürüyorsa, stringlerin eşit olduğunu yazdırdım. Eğer negatif bir değer döndürüyorsa, ilk string alfabetik olarak ikinci stringden önce geliyordu. Eğer pozitif bir değer döndürüyorsa, tam tersi bir durum vardı. Bu soruyla **strcmp** fonksiyonunun çalışma mantığını daha iyi kavradım. **strcmp**, ASCII değerlerine göre karşılaştırma yapıyor, yani küçük harfler büyük harflerden farklı değerlere sahip olabiliyor. Eğer duyarsız bir karşılaştırma yapmak istersem **strcasecmp** gibi bir fonksiyon kullanmam gerektiğini öğrendim.

4.Soru



Bu soruda, kullanıcının girdiği bir stringi yeni bir bellek bölgesine kopyalayarak hem orijinal string hem de kopyanın içeriğini aynı, ancak bellek adreslerini farklı olarak göstermek istedim. Bunun için **malloc** ile yeni bir bellek alanı ayırdım ve **strcpy** ile kopyalama işlemi yaptım. Sonrasında hem orijinal hem de yeni stringin adreslerini yazdırarak gerçekten farklı bölgelerde tutulduklarını gösterdim. Yani **str1** = **str2**; gibi bir şey yaparsam aslında iki değişken de aynı belleği işaret ediyor. Gerçek bir kopya yapmak istiyorsam, **malloc** ile yeni bir alan tahsis edip **strcpy** kullanmam gerekiyor. Aynı zamanda, kullandığım belleği **free**() ile temizlemem gerektiğini unutmadım.

Makefile Dosyası:

