

**AD:** HAFİZE CEMİLE

**SOYAD:** ÖĞÜT

**DERS:** VERİ YAPILARI VE ALGORİTMALAR

**NUMARA:** B210109058

**ÖDEV:**LABİRENT(MAZE) OYUNU:BACKTRAKING UYGULAMASI

**AKADEMİSYEN:**Dr.Öğr.Üyesi Selman HIZAL & Arş.Gör Emin GÜNEY

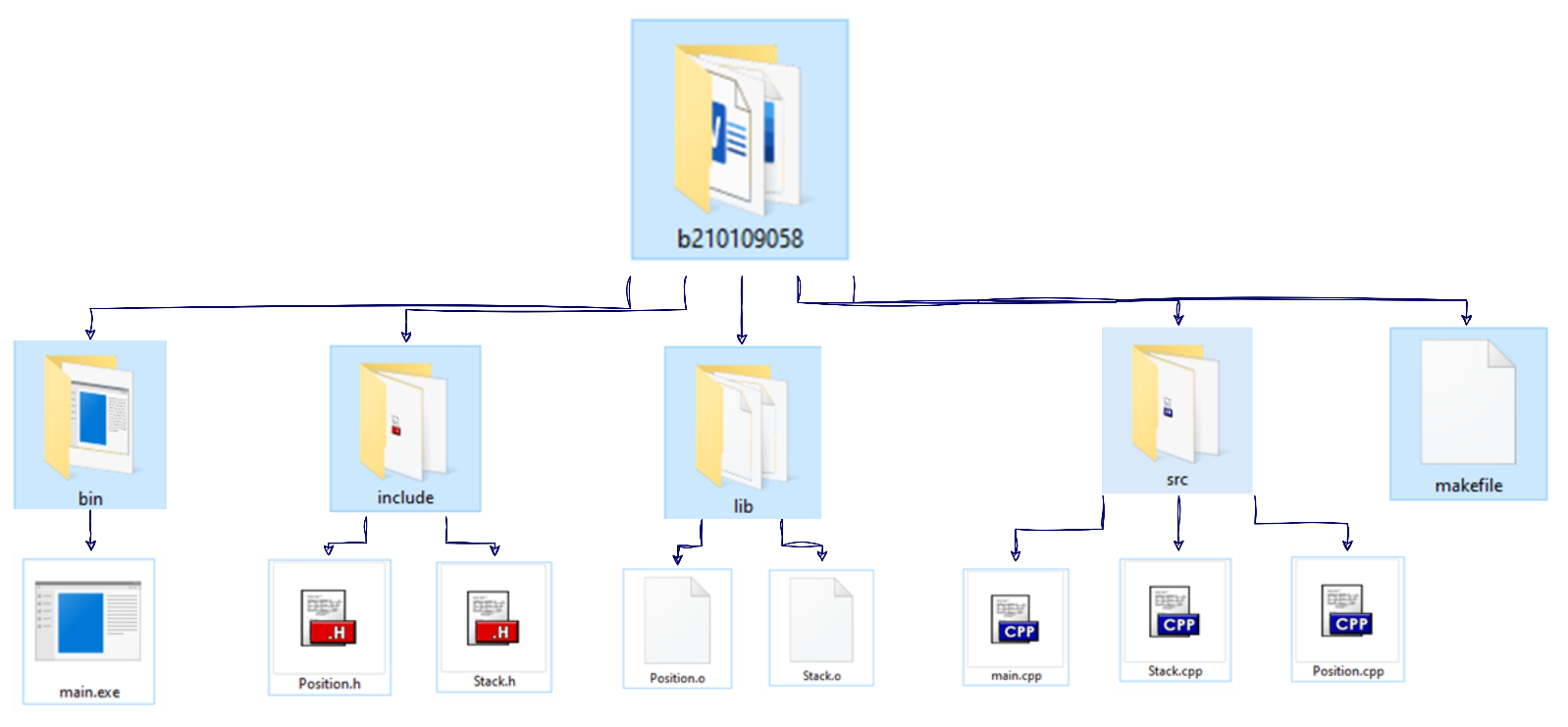
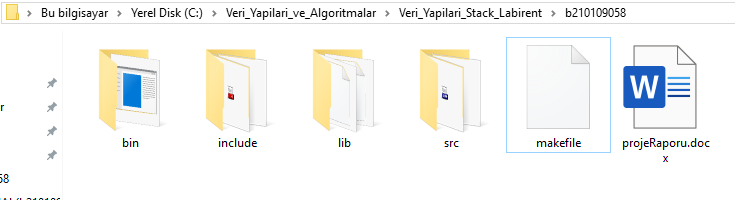
**Ben Kimim?**

Merhabalar, ben Hafize Cemile Öğüt. 15 Mayıs 2000 tarihinde Uşak’ta doğdum. İlk ve orta öğretim hayatımı Uşak’ta tamamladım. Liseyi Uşak Mesleki ve Teknik Anadolu Lisesi, Bilişim /Web Teknolojileri bölümünden mezun oldum.

**Bu Ödevdeki Algoritmam Nedir?**

1. Bir Labirent, blokların N\*M ikili matrisi olarak verilir ve başlangıçta (0,0) yani bir \* simgesi vardır.
2. Labirent [0][0] ve \* labirentteki belirli bir blokta bulunan yiyecekleri yemek ister.
3. Bir labirent matrisinde #, bloğun bir çıkmaz sokak olduğu anlamına gelir ve boşluk karakterleri, bloğun kaynaktan hedefe giden yolda kullanılabileceği anlamına gelir.
4. Karakteri, bloğun çıkmaz bir yol olmaması koşuluyla herhangi bir bloğa herhangi bir yönde (çapraz olarak değil) hareket edebilir.
5. Görev, \* karakterini yiyeceğe ulaşıp ulaşmayacağı herhangi bir yol olup olmadığını kontrol etmektir.
6. Yolu yazdırmak için gerekli değildir.
7. Başlangıçta gerekli dizinlerine sahip yığın yapısı oluşturup, yolları yığına iteceğiz.
8. En üstteki düğümün tüm yönlerine saat yönünün tersine bir şekilde birer birer hareket edeceğiz ve her seferinde yeni bir yol denediğimizde bu düğümü (labirentin bloğu) yığına iteceğiz.
9. Her seferinde en üstteki düğümün gerekli değişkenini arttıracağız, böylece tüm yönler araştırılmadıkça her seferinde yeni bir yön deneyebiliriz.
10. Ayrıca, labirenti hangi bloklarının yolda zaten kullanıldığını veya başka bir deyişle yığında mevcut olduğunu koruyacak ziyaret edilen bir matrisi de koruyacağız.
11. Herhangi bir yönü denerken, labirentin bloğunun çıkmaz bir yol olup olmadığını ve labirentin dışında olup olmadığını da kontrol edeceğiz.
12. Bunu, ya en üstteki düğüm koordinatları yiyeceğin koordinatlarına eşit hale gelirken yapacağız, bu da yiyeceğe ulaştığımız anlamına gelir ya da yığın boşalır, bu da yiyeceğe ulaşmanın olası bir yolu olmadığı anlamına gelir.

**PROGRAM TANITIMI:**

Öncelikle LMS sistemine yükleyeceğim dosyayı göstermek istiyorum. 

Resimde de gördüğünüz gibi, bin, include,lib,src adlarınla yeni klasörler oluşturdum.

Src klasörümde cpp uzantılı main, pointer ve stack dosyalarım var.

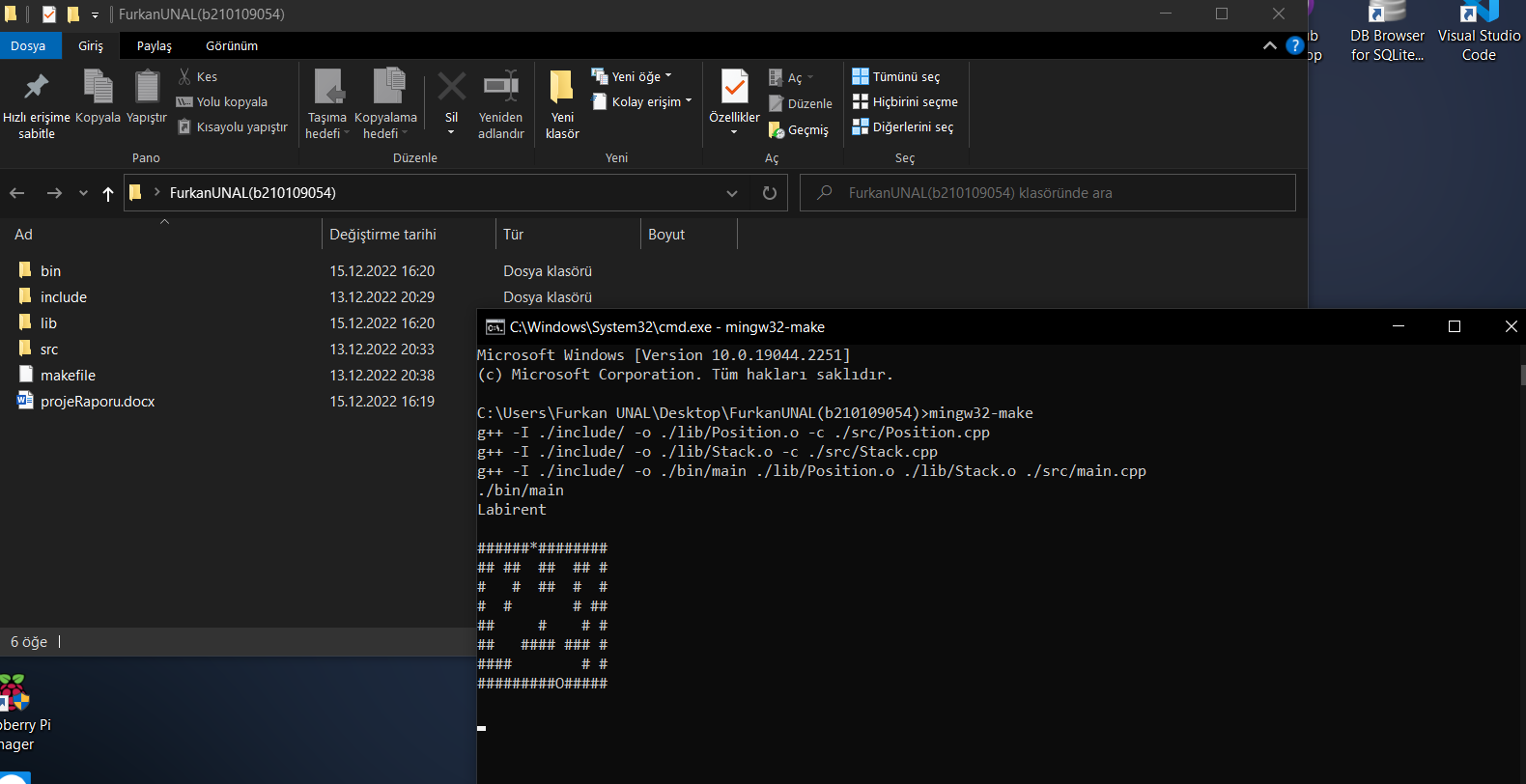
İnclude klasörümde ise başlık uzantılı (.h) dosyalarım var(Position.h, Stack.h).

Daha sonra, makefile dosyamı vsCode açıp gerekli komut satırlarını yazdım;

metin, ekran görüntüsü, yazılım, ekran, görüntüleme içeren bir resim

Açıklama otomatik olarak oluşturuldu

Ve gelelim programın çalışmasına, gerekli komutlarla (mingw32-make) cmd ekranında programımı çalıştırıyorum

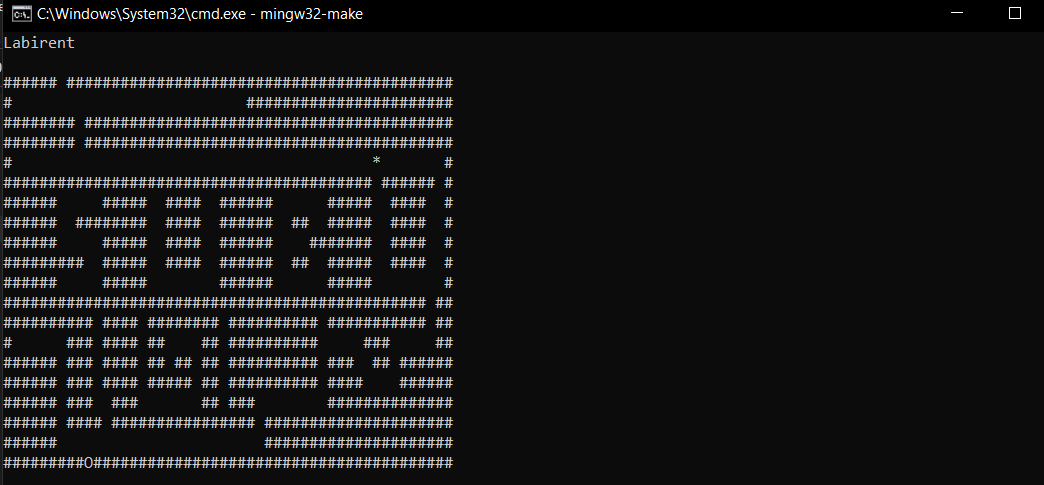


Ve programımın çalışma videosuna da aşağıdaki dosyaya çift tıklayarak izleyebilirsiniz.



Tabii ki yalnızca benim yaptığım sade haritayla kalmayıp, kaynak (main) kodunun içinde gerekli labirent haritanın satır ve sütun sayılarını değiştirerek kendi haritanızı oluşturabilirsiniz. Örnekte kendi girdiğim bir harita var ve labirent bunu çözüyor😊

Kaynak kodlarımda bu haritaya erişebilirsiniz.



**FAYDALANDIĞIM KAYNAKLAR:**

[www.stackoverflow.com](http://www.stackoverflow.com)

[www.geeksforgeeks.org](http://www.geeksforgeeks.org)

[SteverNote | Blog of Tony Stever](https://stevernote.com/)

[https://cplusplus.com](https://cplusplus.com/)

[www.chegg.com](http://www.chegg.com)

[www.github.com](http://www.github.com)

[www.daniweb.com](http://www.daniweb.com)