**LABİTENT OLUŞTURMA**

ERKAN CEYLAN ÜMİT AKSOYLU

**BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**

**KOCAELİ ÜNİVERSİTESİ**

[erkanceylan119@hotmail.com](mailto:mirsaddacic@hotmail.com) [muhammedaksoylu98@gmail.com](mailto:agahsaribas@gmail.com)

**ÖZET**

Projemizi özetlemek gerekirse, projemizin başlangıcında ilerleyen işlemlerimizi de gerçekleştirebilmemiz için öncelikle matrisin değerlerini bağlı liste oluşturarak rastgele deger atatık.Bunları koordinat duzlemindeki gibi x ve y olarak ayırdık.Daha sonra bunları listeye ekledik.Listeye eklediklerimizi de daha sonra yazdırdık.

Devamında ise kullanıcıdan aldığımız labirentin başlangıç ve bitiş noktalarını kullanmak üzere Matriste Gez Fonksiyonunda recursive fonksiyonu kullanarak bir sonraki ya da önceki değerlere ulaştık.Ulaştığımız bu değerleri push ya da pop ederek çıkış noktasına eriştik.

Push ekleyerek çıkmaz bir yolla karşılaştığımızda Geri İzlenim(Backtracking) Algoritmasını kullanarak push ettiklemizi pop ederek yol ayrımından devam edip sonuca ulaştık.

Projemizin sonunda ise ilk önce matrisimizin tamamına 0 değerini atadık.Çıkış yoluna erişmek için gittiğimiz yoldan geçen yerleri 1 olarak değiştirdik.Böylece çıkış yolumuzu bulmuş olduk.

**GİRİŞ**

Projemizin amacı C Programlama dilinde daha önce öğrenmiş olduğumuz bilgilerden olan Bağlı Listeler ve Yığın konularını kullanarak kullanıcıdan matrisin boyutuyla giriş ve çıkış noktalarını kullanıcıdan isteyerek bir labirentin çıkışına ulaşmayı başardık..

Projemiz tamamıyla doğru çalışmakta birlikte bize gereken işlemlerde push ettiklerini ekrana yazdırarak ve sadece çıkış yolunun değerlerini 1 yaparak matrisin çıkışını gösterdik.

**TEMEL BİLGİLER**

Projeyi yaparken kullandığımız teknolojik araçları anlatılacak olursak çok temel iki araçtan yararlandık.

1. Code::Blocks: Özgür açık kaynak kodlu bir C [tümleşik geliştirme ortamıdır](https://tr.0wikipedia.org/index.php?q=aHR0cHM6Ly90ci53aWtpcGVkaWEub3JnL3dpa2kvVCVDMyVCQ21sZSVDNSU5RmlrX2dlbGklQzUlOUZ0aXJtZV9vcnRhbSVDNCVCMQ). [wxWidgets](https://tr.0wikipedia.org/index.php?q=aHR0cHM6Ly90ci53aWtpcGVkaWEub3JnL3dpa2kvV3hXaWRnZXRz" \o "WxWidgets) tabanlı tamamen özelleştirilebilir arabirimiyle, [GNU/Linux](https://tr.0wikipedia.org/index.php?q=aHR0cHM6Ly90ci53aWtpcGVkaWEub3JnL3dpa2kvR05VL0xpbnV4), [Microsoft](https://tr.0wikipedia.org/index.php?q=aHR0cHM6Ly90ci53aWtpcGVkaWEub3JnL3dpa2kvTWljcm9zb2Z0) [Windows](https://tr.0wikipedia.org/index.php?q=aHR0cHM6Ly90ci53aWtpcGVkaWEub3JnL3dpa2kvV2luZG93cw), [MacOS](https://tr.0wikipedia.org/index.php?q=aHR0cHM6Ly90ci53aWtpcGVkaWEub3JnL3dpa2kvTWFjT1M) platformlarında sorunsuzca kullanılabilmektedir. Gelişmiş plugin desteğiyle kod yazımı esnasında ihtiyaç duyabileceğiniz birçok yardımcı fonksiyon sunar ve kod üzerinde tam bir hakimiyet kurmanıza olanak tanır.
2. Google Chrome:Google tarafından geliştirilen bir ağ tarayıcısıdır.Adını çizgisel kullanıcı arayüzünden almaktadır.Microsoft Windows için geliştirilen Beta sürümü ilk kez 2 Eylül 2008 tarihinde kullanıma sunulmuştur.Programlama dili ise C++'dır.

2

[Kenar çubukları, metninizdeki önemli noktaları belirtmek veya hızlı başvuru amacıyla ek bilgiler yerleştirmek (zamanlama gibi) için çok kullanışlıdır.

Genellikle sayfanın soluna, sağına, üstüne veya altına yerleştirilirler. Ancak istediğiniz bir konuma kolayca sürükleyebilirsiniz.

İçeriğinizi eklemeye hazır olduğunuzda, buraya tıklayarak yazmaya başlayabilirsiniz.]

**PROJENİN BÖLÜMLERİ**

**Matrisi Yaratma Fonksiyonu**: İlk olarak matrisYarat adında bir fonksiyon oluşturduk,bu fonksiyonda öncelikle x, y,sayaç adında değerler tanımladık.Daha sonrasında ise Deger adında int tanımlayıp, 1 ile 0 arasında rastgele değerler ürettik.Deger'i listeyeEkle adındaki fonksiyona parametre olarak gönderdik.Daha sonra ise y matrisin boyutundan buyukse eger bir alt satıra geçmek için ve y yi kullandık.Sayaçla ise bunu her seferinde kullanmamızı sagladık.

Not:Matris yaratma işlemi baglı liste ile yapılmıştır.Döngü ile yapılmamıştır.

**Listeye Ekleme Fonksiyonu:**

Fonksiyonun dışında Bağlı Liste oluşturduk ve yeni eleman ekleme işlemi yapmak için hafızadan yer aldık.

Bağlı Listeyi kullanarak yeni elemanın değerini matris yaratma fonksiyonundan gelen değere eşitledik.

Aynı şekilde matris yaratma fonksiyonundan gelen x ve y yi de bağlı liste yardımıyla x\_pozisyonu ve y\_pozisyonu na eşitledik ve listede daha önceden hiç eleman olup olmamasına göre listeye bu yeni elemanı ekledik.

**Listeyi Yazdırma Fonksiyonu:**

Oluşturduğumuz bağlı listeyi temp adında geçici bir değişkene attık. Temp i listenin sonraki değerlerine ulaşmak için oluşturduk.

Önceki fonksiyondan aldığımız listenin değerini yazdırdık.Şayet önceki fonksiyondan aldığımız y\_pozisyonu 0 a eşit olması durumunda alt satıra geçerek matris boyutu şeklinde yazdırmayı sağladık.listenin değerini yazarak da x\_pozisyonunu yazdırmayı başardık.

Not:Buraya kadar matrisi bağlı liste şeklinde oluşturup yazdırılması sağlanmıştır.

**Yolu Bulma Fonksiyonu:**

Parametreler yardımıyla Matris Değerlendir Fonksiyonunun geri döndürülmesidir.

**Matrisi Degerlendirme Fonksiyonu:**

Matriste Gez Fonksiyonunun 0 olması durumunda bağlantı bulunamaması yazmasıdır. 0 olmaması durumunda ise Sonucun Yazdırılması Fonksiyonunun çağrılmasıdır.

**Matriste Gez Fonksiyonu:**

Değeri Kontrol Et Fonksiyonunun 1 olması durumunda yığına eleman eklenmesi ve recursive fonksiyon özelliğini kullanarak Matriste Gez Fonksiyonunun sağa,sola, aşağıya ya da yukarıya gitmesini parametreler yardımıyla x ve y değerlerini kullanarak 1 artırma yada 1 azaltılmasıdır.Eğer bu 4 yöne de gidişi sağlanmazsa yani çıkmaz sokak durumunda yığından eleman çıkarılmasıdır.

**Değeri Kontrol Et Fonksiyonu:**

Değeri Getir Fonksiyonunun 1 olması durumunda ve x ile y değerlerinin 0 dan büyük eşit matrisin elemanından küçük olması durumunda ve Yığında Var mı Fonksiyonunun olması durumunda 1 olarak döner.

**Değeri Getir Fonksiyonu:**

Bağlı liste temp değeri Liste ye eşitlenir.x\_pozisyonunun x olması durumunda ve y\_pozisyonunun y olması durumunda temp->değeri'nin geri döndürülmesidir.Bunların eşit olmaması durumunda ise temp temp->sonrakine eşitlenir.

**YığındaVar mı Fonksiyonu:**

Fonksiyonun dışında oluşturduğumuz YığınStruct TempYığın yapısı yığın a eşitlenir.TempYığın NULL'a eşit olmadığı boyunca pozisyon\_x pos\_x yapılır ve pozisyon\_y pos\_y yapılır ve x ile y ye eşit olması durumunda 0 olmaması durumunda 1 olarak döner.

**Yığın Eleman Ekle Fonksiyonu:**

Yeni Yığın Elemanı Yarat Fonksiyonundan gelen pos\_x ve pos\_y elemanlarının push edilmesi işlemidir.

**Yeni Yığın Elemanı Yarat Fonksiyonu:**

Önce hafıza alınıp daha sonra da parametreler yardımıyla x\_pozisyonunun x y\_pozisyonunun y olması ile yeni eleman oluşturma kısmıdır.

**Yığın Eleman Çıkar Fonksiyonu:**

Yığının Yığın Boş mu fonksiyonu yardımıyla boş olmaması durumunda çıkarılanların Çıkarılan adlı bir dizide toplanıp döndürülmesidir.

**Yığın Bos mu Fonksiyonu:**

Yığının bos olup olmamasını kontrol eden fonksiyondur.

**Sonucu Yazdır Fonksiyonu:**

Matrisin bütün değerlerine 0 yazdırıp Listeye Ekle Fonksiyonu yardımıyla listeye ekleyen ve daha sonra Listeyi Güncelle Fonksiyonu yardımıyla güncelleyip sadece çıkış yolunu yazdıran fonksiyondur.

**Listeye Güncelleme Fonksiyonu:**

Listeyi listede 1 olup olmamasına göre güncelleyen fonksiyondur.

**SONUÇLAR**

Sonuç olarak programımız doğru çalışmaktadır.