**Eroll Ramaxhik - 05.10.901**

**Selim Agovic - 05.11.797**

**Onur Pala - 05.10.871**

**Burak Ozkirdeniz - 05-12-2003**

**Data Structures**

**Proje 2.A**

**OTOPARK SIMULASYONU**

**Mentor : DOC. DR. AYBARS UGUR**

**1.2.1 Gerceklestirilen Platform ,Dil ve Surum Adi :**

**\*Gerceklestirilen Platform : Visual Studio Ultimate 2013**

\***Dil : C#**

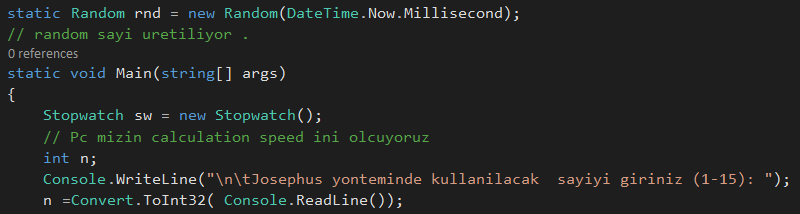
**\*Surum Adi : Proje 2.A**

**1.2.2 Program 2A.1’in Kisa Tanimi :**

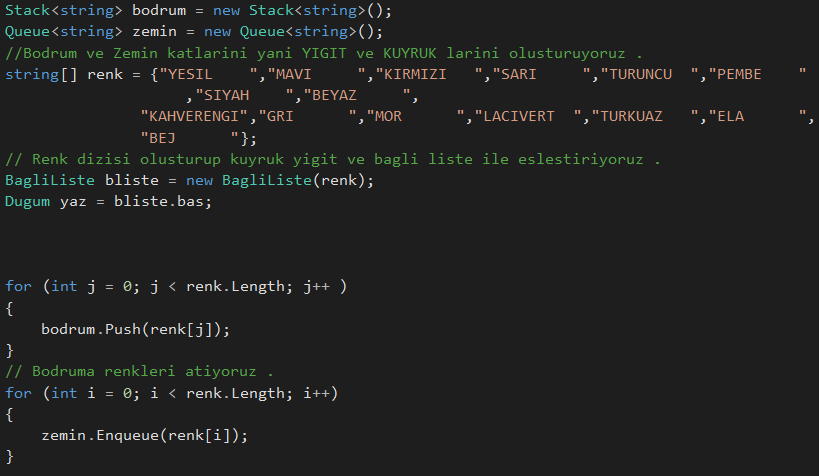
**Bizden istenen 3 katli bir otopark smiluasyonu yapmamiz . En ust kat bagli liste ile yapilacak ve josephus yontemi uygulanacaktir . Josephus yontemi ile kullanicinin girdigi n sayisina gore atlayip secilen arabayi bagli listeden cikarip bir alt kata dahil edecektir ve bu islem bagli liste bos olana kadar devam edecektir ancak cikardigi arabanin bir sonraki arabadan islem devam edecektir yani her cikarma isleminde basa donme olayi olmayacaktir . Orta yani zemin katta kuyruk yapisi olacaktir ve FIFO yontemi ile yani FIRST IN FIRST OUT yontemi ile arabalar cikarilacaktir ,ancak her cikarilan arac isleminden sonra %50 sans ile ya ust kat ya da alt kattan 1 araba kuyrugun sonuna dahil edilecektir . En alt kat yani bodrum da ise araclar YIGIT seklinde tutulacaktir ve LIFO yani LAST IN FIRST OUT uygulanacaktir . Otoparkimiz maximum 45 arac almaktadir ve kat basina dusen arac sayisi 15 tir . Araclar bittiginde program sona erecektir .**

**1.2.3 Program 2A.1 de Kullanilan Veri Yapilari ,Sinif ve Metotlar :**

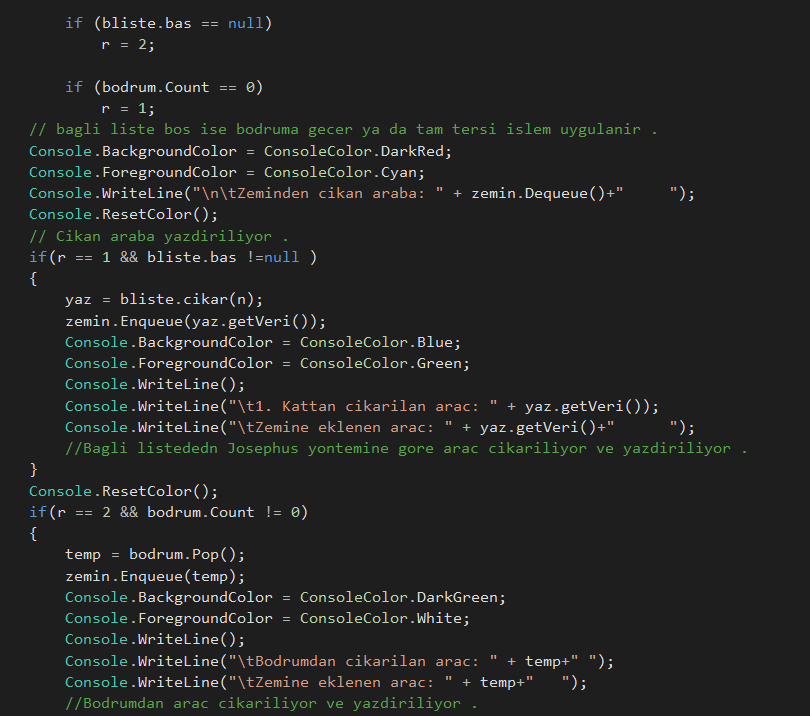
**Program Console seklinde yazilmistir .Ilk once random sayi belirleyerek ise basliyoruz .Islem hizini hesaplamak icin ayriyetten stopwatch hazir fonksiyonunu kullaniyoruz ve ardindan Bagli listede kullanilacak olan sayiyi kullanicidan aliyoruz (Josephus Sayisi).**

****

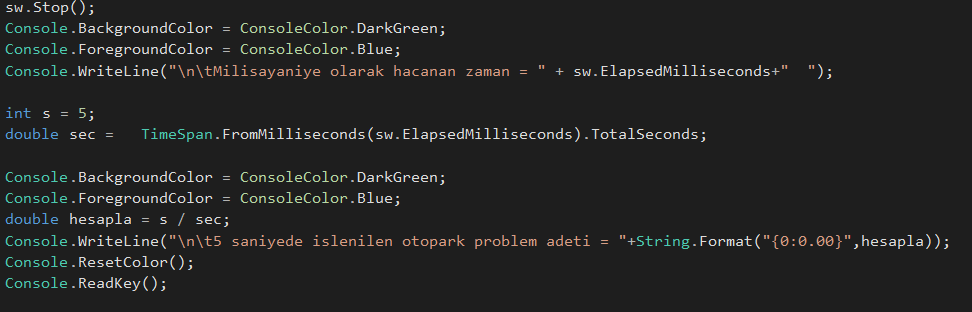
**Bundan sonra ise bodrum yigitini ve zemin kuyrugunu olusturuyoruz ve ardindan butun veri yapilarina gonderecegimiz renk dizisini olusturuyoruz . Bundan sonra ise bodrum yigitina ve zemin kuyruguna renkleri teker teker ekliyoruz .**

****

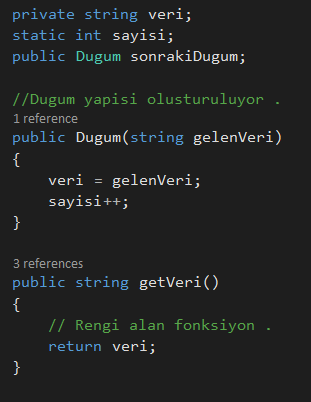
**En basta zemin katin bos olup olmadigini kontrol ediyoruz cunku bos ise araba cikarilmyacaktir yani butun arabalar cikarilmis olacaktir ,degilse o zaman dongunun icine giriyor ve rastgele 2 sayidan yani 1 ve 2 uretiyor ve buna gore hangi fonksiyona girmesi gerektigine karar veriyor .Burda amac zeminden cikarilan arabanin yerine bos kalan yeri doldurmak yani ust kat ve ya alt kattan 1 araba alinacaktir ancak buna random sayi karar vericektir . Eger ust kattaki butun arabalar cikmis ise random sayi sadece alt kattaki donguye girmesi icin uygun sayi uretecektir eger alttaki kat bos ise o zaman tam tersi islem uygulanacaktir .Ust kattan Josephus yontemine gore n sayisi kadar atlayarak arabayi cikaracaktir .Altta ise normnal FIFO yontemi ile araba cikarilacaktir .**

****

**Burada ise programimizin ortalama 5 saniyede kac islem yapabildigini hesapliyoruz ve tabiki bu durum bilgisayarimizin hizina GPU suna da bagli .**

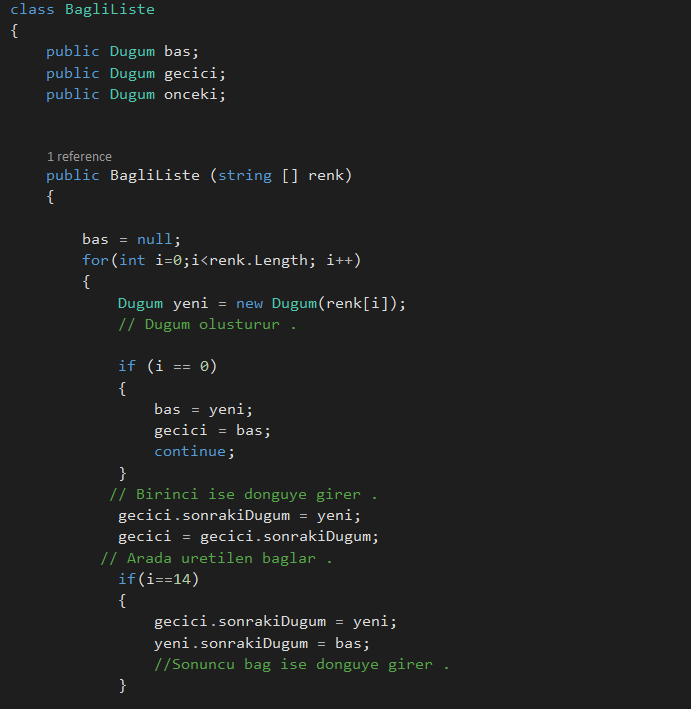
****

**Programimizda ayriyetten bir dugum class i da mevcut . Burada gelen veriyi veri ye esitleyip her gelen veri icin sayaci arttiriyoruz ve altta da arabanin rengini geri donduren fonksiyon mevcut .**

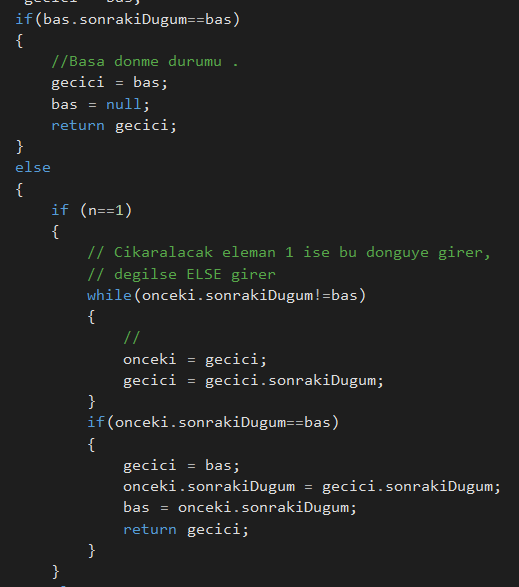
****

**Asagda da belirtildigi gibi getsayisi fonksiyonu arabanin nosunu geri donduruyor . azaltsayi fonksiyonu ise cikarilan arabaya gore sayisini azaltiyor ve en son fonksiyonumuz ise arabanin rengini yazdiriyor .**

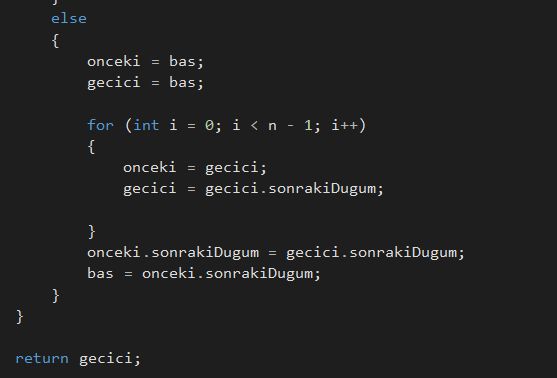
****

**Asagda ayriyetten Bagli liste classimiz mevcut .Bagli liste classimizda bas onceki ve sonraki pointer lari var .Burda yaptigimiz islem teker teker bagli liste uretmek ve verileri icine atmak .**

**Eger basa donme durumu olursa gecici eleman dondurulur . Eger n birinci eleman ise alttaki donguye girer ve eleman cikarilir .**

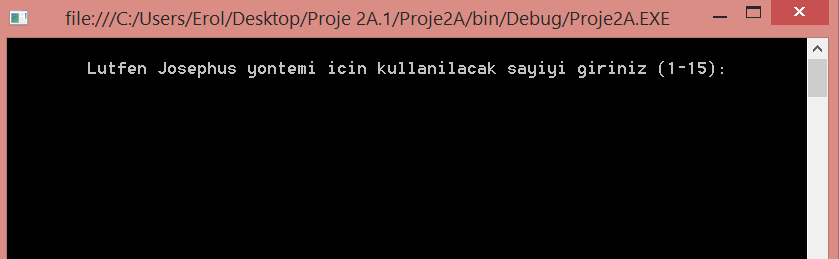
****

**Eger yukarda gosterdigimiz sekildeki donguye girmez ise o zaman else ile siradaki donguye girerek birinci elemandan farkli yani bas elemanindan sonra gelen elemanlarin cikarma islemi uygulanir .**

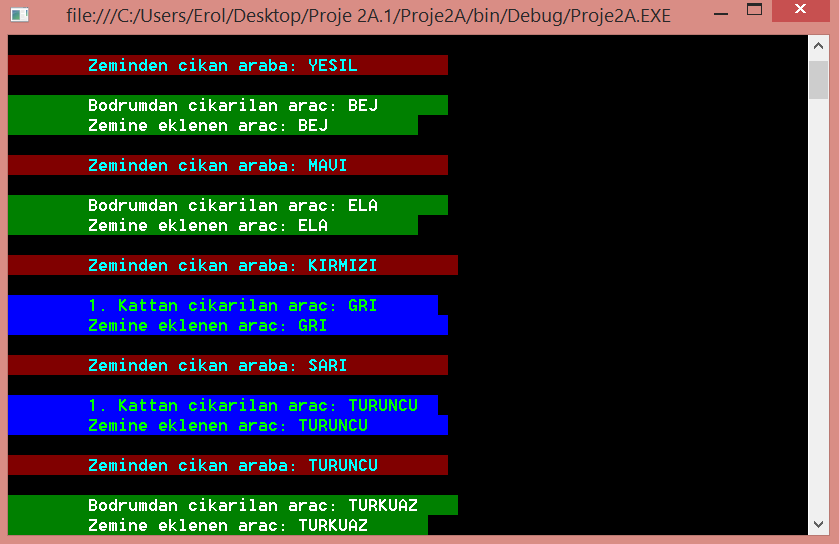
****

**1.2.4 Program 2A.1 de Elde Edilen Ornek Sonuclar :**

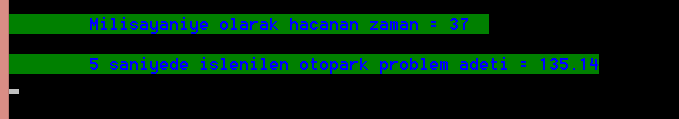
**Kullanicinda Bagli listede kullanilacak olan sayiyi istiyoruz yani bagli listede kac kez atlama yapacagina dair sayiyi kullanicidan aliyoruz (Josephus Yontemi).**

****

**Programimiz araclari teker teker zemin kattan cikariyor ve cikarilan araclarin yerine ust kattan (1.KAT) ve ya alt kattan ekleme yapiyor (BODRUM).Program random sayisina gore yani %50 sansa gore ya yukardan arac cikariyor ya da asagdan .**

****

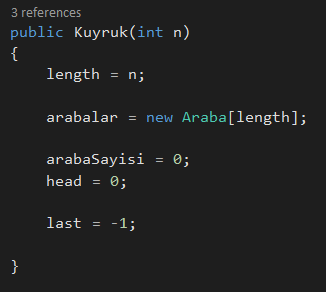
**Bu ise programimizin 5 saniyede ortalama kac otopark simulasyon hesaplamasi yapabilecegini gosteren cikti .**

****

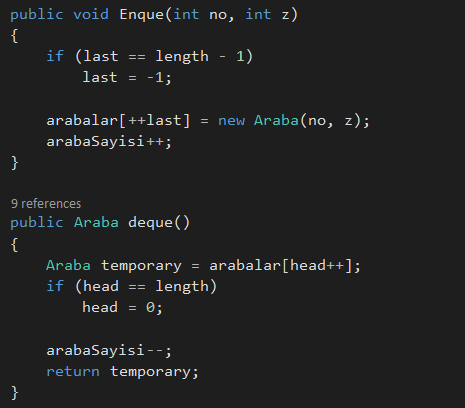
**1.2.5 Program 2A.2’in Kisa Tanimi :**

**2. kisimda bizden istenen yukardaki programa benzer ancak sadece kuyruk yapisi ile tasarlanmasi . Bu sefer her bir aracin random 10-300 saniye arasinda bir sayisi olacak ve en ondeki arac cikmadan baska arac cikamayacaktir .Buna gore FIFO kullanilacaktir ve her bir aracin random zaman suresi ve numarasi olacaktir .**

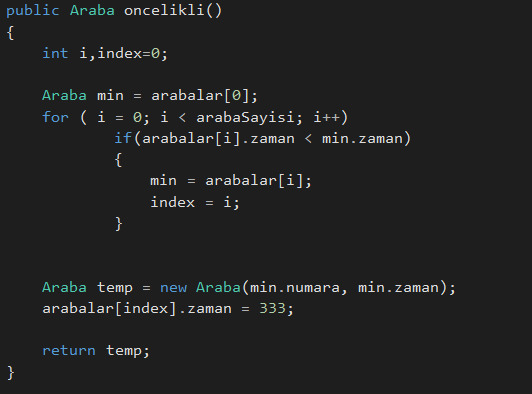
**1.2.3 Program 2A.2 de Kullanilan Veri Yapilari ,Sinif ve Metotlar :**

**Gonderilen n sayisina gore kuyrugun uzunlugu belirlenecek ve o uzunlukta bir kuyruk uretilecektir .**

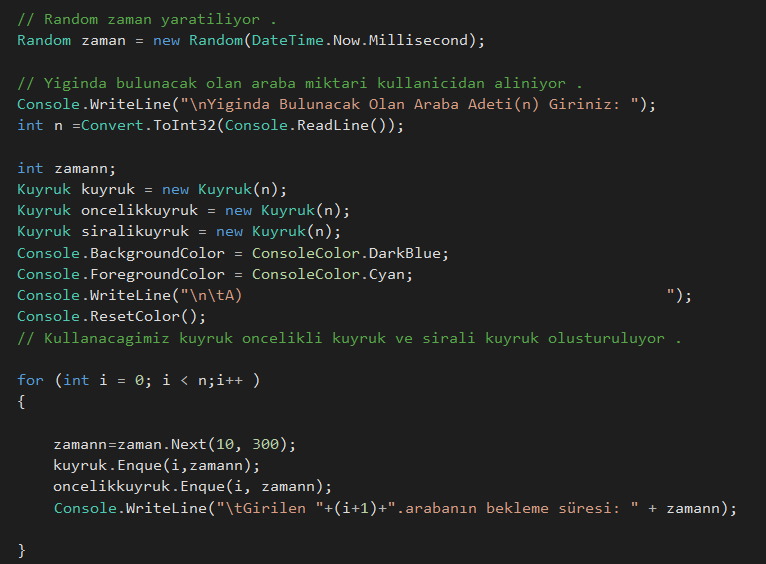
**Kuyrukta extradan ekleme ve silme metotlari da mevcut .**

****

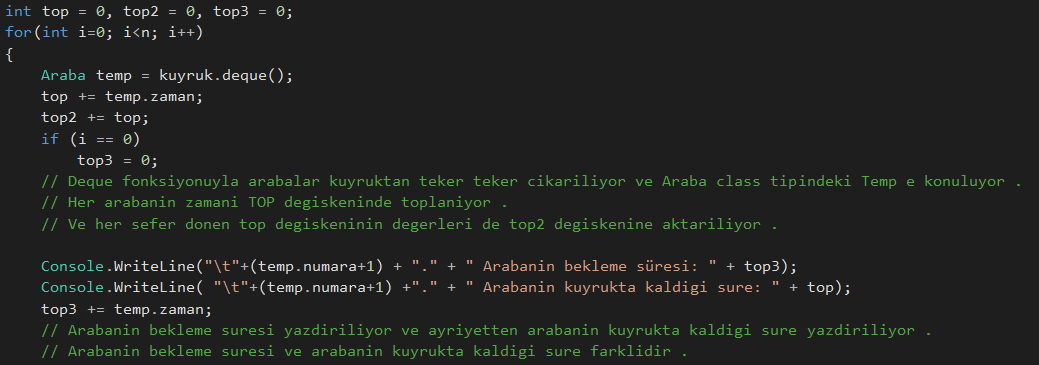
**Oncelikli kuyruk metotunda bizler sadece sirali cikarma yapiyoruz yani kucukten buyugue dogru sirali cikarma islemi uyguluyoruz ancak icinde kucukten buyuge sirali bicimde degil . Cikardigimiz verileri baska bir kuyruk yapisina sirali olarak ekliyoruz (siralikuyruk).**

****

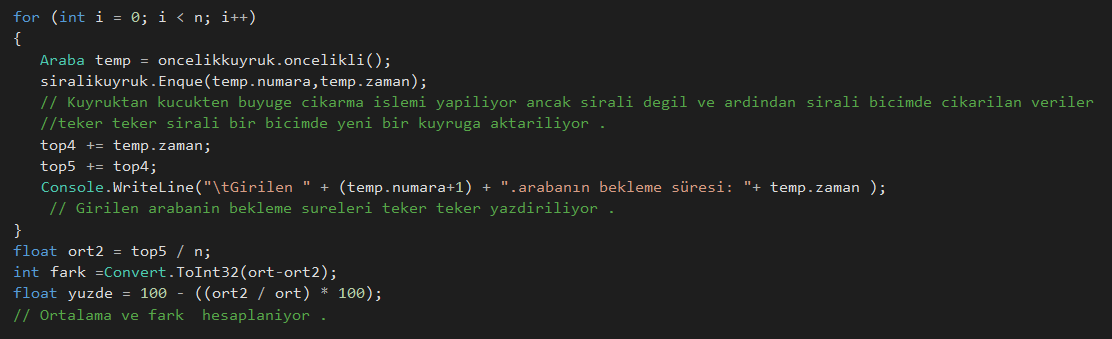
**Burada random zaman tanimliyoruz ve sonradan n sayisini yani kuyrukta bulunacak olan n sayisini kullanicidan aliyoruz .3 tane kuyruk olusturuyoruz .Ondan sonra 10-300 arasinda uretilen saniye ve numarasina gore kuyruga ekleme islemi yapip ekranda eklenen verinin bilgilerini yazdiriyoruz .**

****

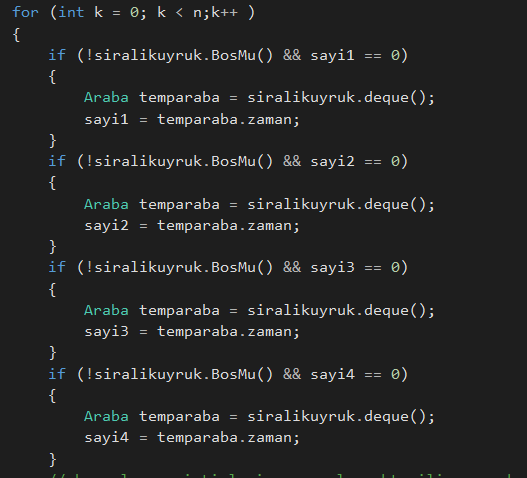
**Sil fonksiyonu ile arabalar kuyruktan teker teker cikartiliyor ve Araba class tipindeki Temp e konuluyor. Her arabanin islem suresi TOP degiskeninde toplaniyor . Ve her sefer donen top degiskeninin degerleri TOP2 degiskenine aktariliyor .TOP3 degiskeni isle toplam arabanin kuyrukta kaldigi sureyi hesapliyor .**

****

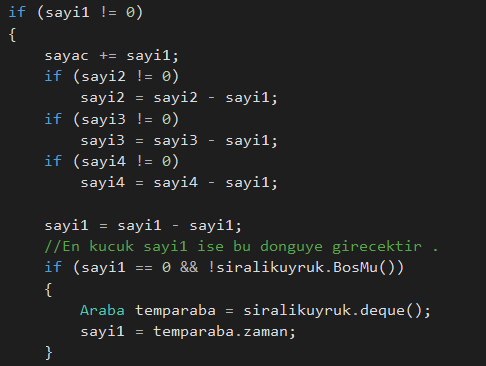
**Onceliklikuyruk ile arabalar kucukten buyuge dogru cikartilarak sirali kuyruga ekleniyor . Cikartilan arabanin bekleme suresi ve numarasi yazdiriliyor ve ayriyetten diger cikartma islemine gore fark ve yuzdesi hesaplaniyor ve dongu bitiminde yazdiriliyor .**

****

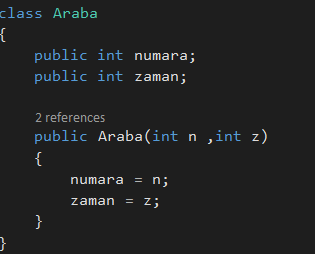
**Son islem ise arabalarin 4 cikisli bir otoparktan cikartilmasi . Arabalar kucukten buyuge dogru sirali olarak cikartilacak ve 4 tane degiskene aktarilacak . En kucuk zamanli veri 1. Degiskene gittigi icin cikarma islemi 1. Degiskenden baslayacak ve sirali olarak devam edecek . Cikarilan arabanin zamani diger degiskenlerdeki zamanlardan eksiltilecek ve bosalan degiskenin yerine kuyruktan tekrar araba gonderilecek . Bu islem kuyruktaki arabalar bitene kadar devam edecek . 4 cikisli bir otopark yaptigimiz diger tum otoparklara nazaran daha hizli islem yapacak ve zamani minimuma indirecektir .**

****

**Bu ise sayinin kucuk olup olmadigini bulan fonksiyonumuz (sayi1 icin gecerli).**

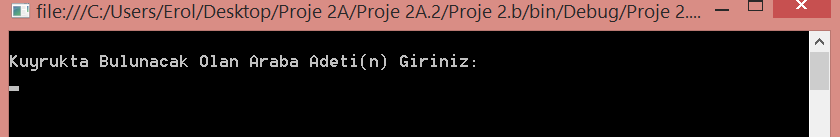
****

**Bu ise kullandigimiz class yapisi . Icinde numara ve zaman iceriyor .**

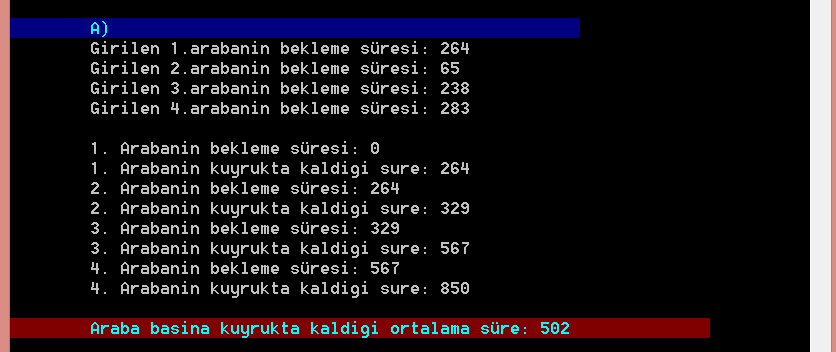
****

**1.2.4 Program 2A.1 de Elde Edilen Ornek Sonuclar :**

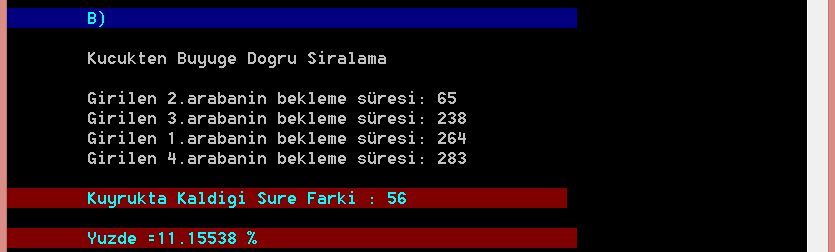
**En basta kullanicidan kac tane arac istedigini soruyoruz ve kullanicinin girmek istedigi arac sayisini aliyoruz .**

****

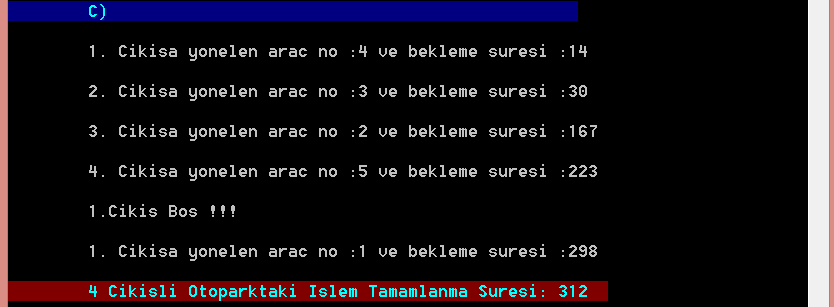
**Ornek olarak biz 5 sayi girdik yani 5 adet arac uretilecek ve kuyruga yerlestirilecek. Araba suresi ve kuyruk suresi yazdirilacak .**

****

**Burada kucukten buyuge dogru yazdiriyoruz ve A sikkina gore fark ve yuzdeyi hesapliyoruz . Yani sirali degilde karisik sekilde arabalari cikarirsak zaman uzayacak ,ancak sirali bir sekilde cikarirsak zamandan kazanabiliriz .**

****

**Burada 4 cikisa yonelen araclar gozukuyor ve herhangi bir cikis bos kaldiginda yerine arac ekleniyor .En sonda da islem tamamlanma suresi yazdiriliyor .**

****

**1.2.5 Yazilim Gelistirme Icin Harcanan Sureler :**

**Eroll Ramaxhik : Toplam (3h)**

**Selim Agovic : Toplam(3h)**

**Onur Pala : Toplam (3h)**

**Burak Ozkirdeniz : Toplam (3h)**

**Grup : Toplam (2h)**