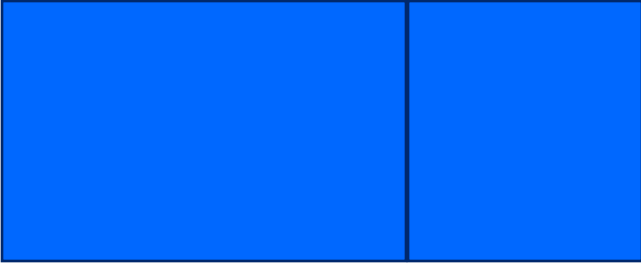


DGH BİTİRME PROJESİ

ALGORİTMA ÖDEVLERİ SUNUM

Algoritma Ödevi Soru-1

Çevreden Alana



Dikdörtgenin çevre uzunluğu ve karenin tek kenar uzunluğu kullanıcı tarafından giriliyor. Buna göre dikdörtgenin alanını hesaplayan programı yazınız.

Algortima Ödevi Soru-1 Çözümü

```
static void Main(string[] args)
```

```
{
```

```
    int uzunKenar, cevre, alan, kenar;
```

```
    Console.Write("Dikdörtgenin çevre uzunluğunu girin:");
```

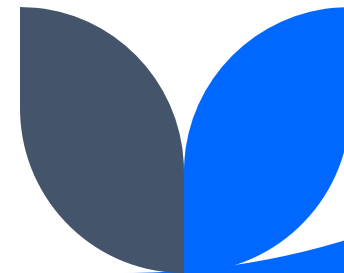
```
    cevre = Convert.ToInt32(Console.ReadLine());
```

```
    Console.Write("Karenin tek kenar uzunluğunu girin:");
```

```
    kenar = Convert.ToInt32(Console.ReadLine());
```

```
    uzunKenar = (cevre - (2 * kenar)) / 2;
```

```
    alan = uzunKenar * kenar;
```



Soru-1 Çözüm devamı

```
Console.WriteLine("Dikdörtgenin alanı : " + alan);
```

```
}
```

Algoritma Ödevi Soru-2

Köşegenleri yer değiştir

1					10
	2			20	
		3	30		
		40	4		
	50			5	
60					6

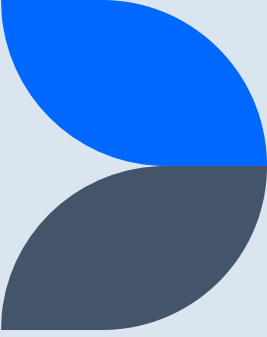
10					1
	20			2	
		30	3		
		4	40		
	5			50	
6					60

Algoritma Ödevi Soru-2

Girilen N değerine göre $N \times N$ boyutlu bir matrisin hücrelerine 0-100 aralığında rastgele bir sayı yerleştiriniz. Daha sonra iki köşegeninde bulunan sayıların yerlerini aşağıdaki gibi ters çeviren programı yazınız

Algoritma Ödevi Soru-2 Çözümü

```
static void Main(string[] args)
{
    Console.Write("Matris boyutunu giriniz: ");
    int n = Convert.ToInt32(Console.ReadLine());
    int[,] matris = new int[n, n];
    Random rand = new Random();
    for (int i = 0; i < n; i++)
```

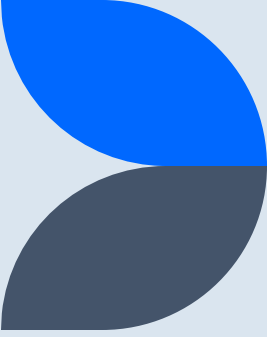


```
for (int j = 0; j < n; j++)  
{  
    matris[i, j] = rand.Next(0, 101);  
}  
}  
Console.WriteLine("Rastgele oluşturulan matris:");  
for (int i = 0; i < n; i++)  
{  
    for (int j = 0; j < n; j++)  
    {  
        Console.Write(matris[i, j] + " ");  
    }  
    Console.WriteLine();  
}
```

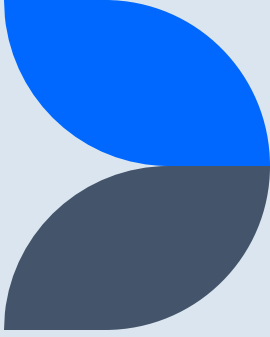


```
}
```

```
Console.WriteLine("Köşegenleri ters çevrilmiş matris:");  
for (int i = 0; i < n; i++)  
{  
    for (int j = 0; j < n; j++)  
    {  
        if (i == j || i + j == n - 1)  
        {  
            Console.Write(matris[n - 1 - i, n - 1 - j] + " ");  
        }  
        else  
        {  
            Console.Write(matris[i, j] + " ");  
        }  
    }  
}
```



Soru-2 Çözüm



- Kullanıcıdan matris boyutu olan n değeri alınır.
- $n \times n$ boyutunda bir matris oluşturulur.
- Her hücreye, 0-100 aralığında rastgele bir sayı atanır.
- Matris, ekrana yazdırılır.
- Matrisin iki köşegeninde bulunan sayıların yerleri ters çevrilir.
- Sonuç matrisi, ekrana yazdırılır.

Soru-2 Çözüm

İlk `if` bloğu, matrisin ana köşegenindeki elemanları ters çevirirken, ikinci `if` bloğu ise matrisin yan köşegenindeki elemanları ters çevirir.

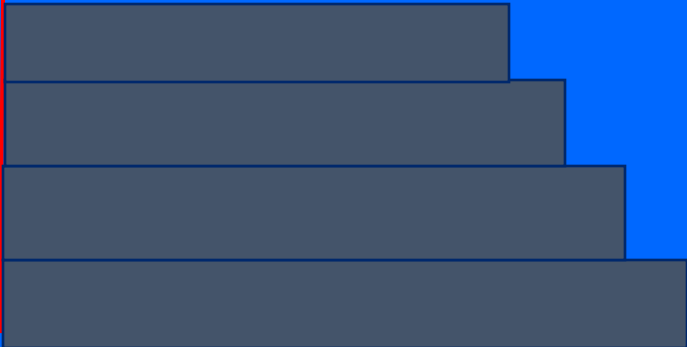
`i == j` ifadesi, ana köşegenin elemanlarını belirler. Bu koşul, `i` ve `j` değişkenlerinin değerlerinin eşit olduğunda doğrudur.

`i + j == n - 1` ifadesi, yan köşegenin elemanlarını belirler. Bu koşul, `i` ve `j` değişkenlerinin toplamının `n - 1` değerine eşit olduğunda doğrudur.

Eğer bir eleman ana veya yan köşegende ise, o elemanın yeri ters çevrilir. Aksi takdirde, elemanın yeri değiştirilmeden kalır.

Algoritma Ödevi Soru-3

$(x+2)$ cm



Kısa kenar x

Her kağıdın uzun kenarı öncekinden 2cm daha uzun n

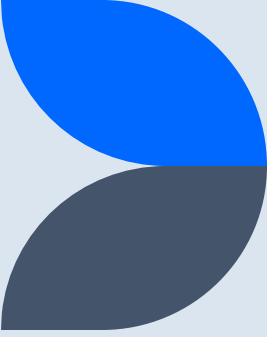
Kağıt alt alta koyuluyor.

Her kağıt koymada şeklin alanını yazınız.

x ve n kullanıcı tarafından girilecek.

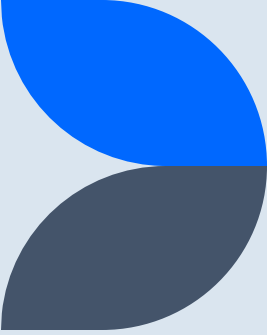
Özyinelemeli çözülecek.

Algoritma Ödevi Soru-3 Çözüm



```
static void Main(string[] args)
{
    Console.Write("Kısa kenar uzunluğunu giriniz: ");
    int x = Convert.ToInt32(Console.ReadLine());
    Console.Write("Kağıt sayısını giriniz: ");
    int n = Convert.ToInt32(Console.ReadLine());
    Console.WriteLine("Toplam alan: " + ToplamAlan(x, n));
}
```

```
public static int ToplamAlan(int x, int n)
{
    if (n == 1)
    {
        return x * (x + 2);
    }
    else
    {
        return ToplamAlan(x, n - 1) + (x + 2 * (n - 1)) * x;
    }
}
```



- Kullanıcıdan kısa kenar uzunluğu olan x ve kağıt sayısı olan n değerleri alınır.
- ToplamAlan metodu, n adet kağıdın alt alta konulmasıyla oluşan şeklin alanını hesaplar. Bu metot, özyinelemeli bir metot olarak çalışır.

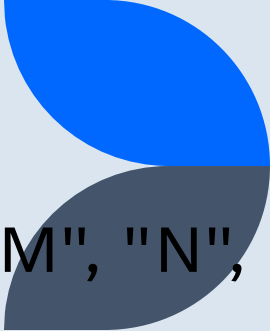
else bloğu, özyinelemeli bir metot kullanarak matrisin alanını hesaplamak için gereklidir. Eğer n değeri 1'e eşitse, yani matris sadece bir hücreden oluşuyorsa, metot $x * (x + 2)$ ifadesini döndürür. Ancak, n değeri 1'den farklıysa, yani matris birden fazla hücreden oluşuyorsa, metot kendisini bir önceki satırdaki $\text{ToplamAlan}(x, n - 1)$ ifadesiyle çağırır ve sonucuna $(x + 2 * (n - 1)) * x$ ifadesini ekler. Bu işlem, matrisin her bir satırının alanını hesaplar ve sonunda toplam alanı döndürür.

Program, hesaplanan alanı ekrana yazdırır.

Algoritma Ödevi Soru-4

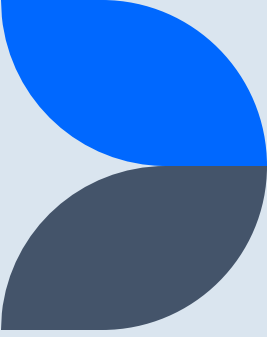
İngiliz alfabesinde İ,Ö,Ü,Ç,Ğ,Ş harfleri yoktur ve bizim alfabemize göre Q,W,X harfleri fazladan vardır. İngiliz alfabesindeki harfleri alfabetik olarak sıralandığında ilk harf sahip olduğu sıraya numarası kadar dosyaya yazılacaktır. Örneğin Z harfi dosyaya 26 defa yazılacaktır , B harfide 2 defaa yazılacaktır, her harf bir satıra yazılacaktır ve ondan sonra yeni satır karakteri yazılacaktır. If else() , else if() , switch-case komutlarını kullanmadan İngiliz alfabesini dosyaya istenilen tarzda yazan programı yazınız

Algoritma Ödevi Soru-4 Çözümü



```
static void Main(string[] args)
{
    string[] alfabe = { "A", "B", "C", "D", "E", "F", "G", "H", "I", "J", "K", "L", "M", "N",
"O", "P", "Q", "R", "S", "T", "U", "V", "W", "X", "Y", "Z" };
    string fileName = "alphabet.txt";
    using (StreamWriter writer = new StreamWriter(fileName))
    {
        for (int i = 0; i < alfabe.Length; i++)
        {
            for (int j = 0; j <= i; j++)
            {
                writer.Write(alfabe[j]);
            }
            writer.WriteLine();
        }
    }
}
```

```
}  
}  
    Console.WriteLine("Dosya başarıyla oluşturuldu.");  
}
```



Bu program, İngiliz alfabesini dosyaya yazmak için `StreamWriter` sınıfını kullanır. Program, `alphabet` dizisindeki harfleri alfabetik olarak sıralar ve her harfin dosyaya yazılması gereken sayısını `counts` dizisinde tutar. Daha sonra, her harf için dosyaya yazılır ve her harfin ardından yeni bir satır karakteri eklenir.

Algoritma Ödevi Soru-5

$$e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots + \frac{x^{10}}{10!}$$

Algoritma Ödevi Soru-5 Çözümü

OOP ile Basit Banka Otomasyonu

- 1) Banka hesabı adında bir sınıf oluşturulurr.
- 2) Banka hesabı sınıfı para yatırma , para çekme ve bakiye görüntüleme özelliklerini içerir.
- 3) Main metodu içine yazılan komutlara göre ilgili banka hesabı sınıfındaki ilgili işlemleri yerine getirir.

```
static void Main(string[] args)
{   BankaHesabi musteri = new BankaHesabi();
```

```
    musteri.bakiye = 1000;
```

```
    Console.Write("Hesaptaki bakiyeniz : ");
```

```
    musteri.bakiyeGoruntule();
```

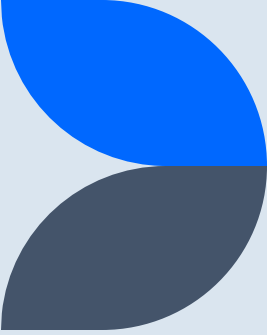
```
    Console.WriteLine("Yapmak istediğiniz işlemi seçin:");
```

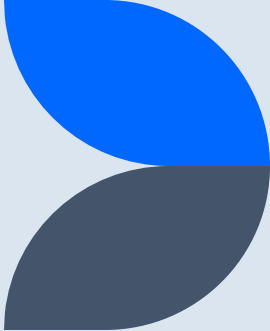
```
    Console.WriteLine("1 - Para yatırma");
```

```
    Console.WriteLine("2 - Para çekme");
```

```
    Console.WriteLine("3 - Bakiye görüntüleme");
```

```
    int secim = Convert.ToInt32(Console.ReadLine());
```





```
switch (secim)
```

```
{
```

```
    case 1:
```

```
        Console.WriteLine("Yatırmak istediğiniz tutarı girin:");
```

```
        int yatirilanMiktar = Convert.ToInt32(Console.ReadLine());
```

```
        musterı.paraYatir(yatirilanMiktar);
```

```
        Console.WriteLine("*****");
```

```
        Console.Write("Güncel Bakiyeniz : ");
```

```
        musterı.bakiyeGoruntule();
```

```
        break;
```

Bu blok, kullanıcıdan yatırmak istediği tutarı alır ve paraYatır metodunu kullanarak hesaba yatırır. paraYatır metodu, hesap bakiyesine belirtilen miktarı ekler. Daha sonra, güncel bakiyeyi görüntülemek için bakiyeGoruntule metodu çağrılır.

case 2:

```
Console.WriteLine("Çekmek istediğiniz tutarı girin:");  
int cekilenMiktar = Convert.ToInt32(Console.ReadLine());  
musteri.paraCek(cekilenMiktar);  
Console.WriteLine("*****");  
Console.Write("Güncel Bakiyeniz : ");  
musteri.bakiyeGoruntule();  
break;
```

Bu kod bloğu, kullanıcının hesaptan para çekme işlemini sağlar. Kullanıcının para çekmek istediği durumda çalıştırılır. Bu blok, kullanıcıdan çekmek istediği tutarı alır ve paraCek metodunu kullanarak hesaptan çeker. paraCek metodu, hesap bakiyesinden belirtilen miktarı çıkarır. Daha sonra, güncel bakiyeyi görüntülemek için bakiyeGoruntule metodu çalışır.

case 3:

```
Console.Write("Güncel Bakiyeniz : ");  
musteri.bakiyeGoruntule();  
break;
```

default:

```
Console.WriteLine("Hatalı seçim yaptınız ! ");  
break;
```

Bu kod bloğu, kullanıcının bakiyesini görüntülemesini sağlar. case 3 bloğu, kullanıcının bakiye görüntülemek istediği durumda çalıştırılır. Bu blok, bakiyeGoruntule metodunu çağırarak hesap bakiyesini görüntüler. break ifadesi, her bir case bloğunun sonunda kullanılır ve bu ifade, switch ifadesinden çıkılmasını sağlar. default bloğu, switch ifadesindeki hiçbir case bloğunun eşleşmediği durumlarda çalıştırılır.

```
public class BankaHesabi
```

```
{
```

```
    public int bakiye;
```

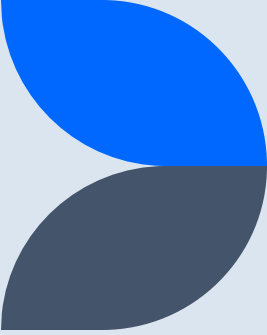
```
    public void paraYatir(int miktar)
```

```
{
```

```
        bakiye += miktar;
```

```
        Console.WriteLine(miktar + " TL hesaba yatırıldı.");
```

```
}
```



```
public void paraCek(int miktar)
```

```
{
```

```
    if (bakiye < miktar)
```

```
    {
```

```
        Console.WriteLine("Yetersiz bakiye. İşlem gerçekleştirilemedi.");
```

```
    }
```

```
    else
```

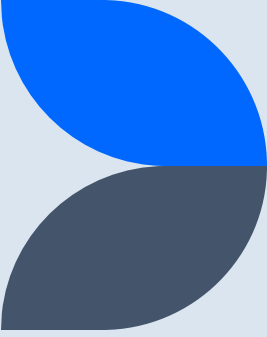
```
    {
```

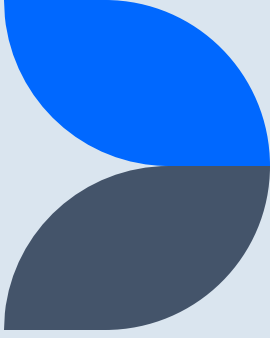
```
        bakiye -= miktar;
```

```
        Console.WriteLine(miktar + " TL hesaptan çekildi.");
```

```
    }
```

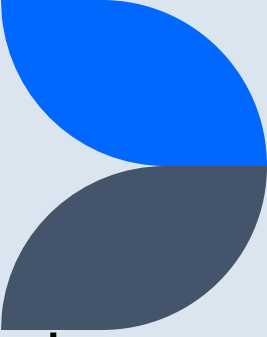
```
}
```





```
public void bakiyeGoruntule()  
{  
    Console.WriteLine(bakiye + " TL");  
}
```

Bu kod bloğu kullanıcının bakiyesini görüntülemesini sağlar.



Bu kod bloğu, `BankaHesabi` adlı bir sınıf içerir. Bu sınıf, bir banka hesabının temel özelliklerini ve işlevlerini tanımlar.

Kod, `paraYatir`, `paraCek` ve `bakiyeGoruntule` adlı üç metod içerir. Bu metodlar, para yatırma, para çekme ve bakiye görüntüleme işlemlerini gerçekleştirir. `bakiye` adlı bir değişken, hesap bakiyesini tutar. `paraYatir` yöntemi, hesap bakiyesine belirtilen miktarı ekler. `paraCek` yöntemi, hesap bakiyesinden belirtilen miktarı çıkarır. `bakiyeGoruntule` yöntemi, hesap bakiyesini görüntüler