

Örnek sorular

Sıralı bir dizide arama yapmak için hazırlanmış bir BinarySearch() fonksiyonu aşağıda verilmiştir. Fonksiyon istenen sayıyı bulursa indisini aksi takdirde -1 döndürmektedir. Binary search işlemini recursive olarak çözen BinarySearchR() fonksiyonu boşlukları doldurarak tamamlayınız

```
int BinarySearch(int arr[], int bas, int son, int x)
{
    while (bas <= son)
    {
        int m = bas + (son-bas)/2;
        if (arr[m] == x) return m;
        if (arr[m] < x) bas = m + 1;
        else son = m - 1;
    }
    return -1;
}
```

```
int BinarySearchR(int arr[], int bas, int son, int x)
{
    }
}
```

Sıralı bir dizide arama yapmak için hazırlanmış bir BinarySearch() fonksiyonu aşağıda verilmiştir. Fonksiyon istenen sayıyı bulursa indisini aksi takdirde -1 döndürmektedir. Binary search işlemini recursive olarak çözen BinarySearchR() fonksiyonu boşlukları doldurarak tamamlayınız

```
int BinarySearch(int arr[], int bas, int son, int x)
{
    while (bas <= son)
    {
        int m = bas + (son-bas)/2;
        if (arr[m] == x)    return m;
        if (arr[m] < x)    bas = m + 1;
        else                son = m - 1;
    }
    return -1;
}
```

```
int BinarySearchR(int arr[], int bas, int son, int x)
{
    if (bas <= son)
    {
        int mid = bas + (son - bas)/2;
        if (arr[mid] == x) return mid;
        if (arr[mid] < x)  return BinarySearchR(arr, mid+1, son, x);
        return BinarySearchR(arr, bas, mid-1, x);
    }
    else
        return -1;
}
```

N*N lik bir oyun alanında nesneler ve en alt satıra yerleştirilmiş bir bomba bulunmaktadır. Nesneler pozitif sayılar ile, boşluklar 0 ile, bomba ise negatif bir sayı ile gösterilmektedir. Bombanın gücü (guc) ilgili negatif sayının mutlak değerine eşittir. Her bomba patladığında, kendisini ve kendisiyle aynı kolonda bulunan nesneleri yok etmektedir. Her bomba üst komşularından gücü kadar nesneyi yok edebilir. Yok edemediği nesneler boşalan hücrelere düşmektedir. $guc < N-1$ olduğuna göre, kullanıcıdan alınan oyun alanındaki bombayı patlatarak oyun alanının son halini ekrana yazdıran programı yazınız.

0	0	0	0	0	6
1	3	0	4	6	5
5	2	0	2	3	4
4	8	1	1	5	3
4	2	2	1	2	2
2	3	4	4	-2	1

Oyun Alanının İlk Hali

0	0	0	0	0	6
1	3	0	4	0	5
5	2	0	2	0	4
4	8	1	1	0	3
4	2	2	1	6	2
2	3	4	4	3	1

Oyun Alanının Son Hali

```
int main()
{
    int mat[6][6]=
        {{0,0,0,0,0,6},
         {1,3,0,4,6,5},
         {5,2,0,2,3,4},
         {4,8,1,1,5,3},
         {4,2,2,1,2,2},
         {2,3,4,4,-2,1}};
    int i,j=0,n=6,guc;
    ??????
    return 0;
}
```

0	0	0	0	0	6
1	3	0	4	6	5
5	2	0	2	3	4
4	8	1	1	5	3
4	2	2	1	2	2
2	3	4	4	-2	1

Oyun Alanının İlk Hali

0	0	0	0	0	6
1	3	0	4	0	5
5	2	0	2	0	4
4	8	1	1	0	3
4	2	2	1	6	2
2	3	4	4	3	1

Oyun Alanının Son Hali

```
int main()
{
    int mat[6][6]=
        {{0,0,0,0,0,6},
         {1,3,0,4,6,5},
         {5,2,0,2,3,4},
         {4,8,1,1,5,3},
         {4,2,2,1,2,2},
         {2,3,4,4,-2,1}};

    int i,j=0,n=6,guc;
    ??????
    return 0;
}
```

```
int main()
{
    int mat[6][6]=
        {{0,0,0,0,0,6},
         {1,3,0,4,6,5},
         {5,2,0,2,3,4},
         {4,8,1,1,5,3},
         {4,2,2,1,2,2},
         {2,3,4,4,-2,1}};

    int i,j=0,n=6,guc;

    while (mat[n-1][j]>0) j++;
    guc= mat[n-1][j]*-1;
    for (i=n-guc-2;i>=0;i--)
        mat[i+guc+1][j]=mat[i][j];
    for (i=0;i<guc+1;i++)
        mat[i][j]=0;

    for (i=0;i<n;i++)
    {
        for (j=0;j<n;j++)
            printf("%d ",mat[i][j]);
        printf("\n");
    }
    return 0;
}
```

Not1: Sadece sonucu yazmanız puan kazandırmaz. Sonuca nasıl ulaştığınızı açıklamamız/göstermeniz gerekmektedir.

Not 2: $i=0:N$ toplam(a^i) = $(a^{N+1}-1)/(a-1)$

Çalışma zamanları aşağıdaki gibi verilmiş özyinelemeli fonksiyonların çalışma zamanını özyineleme ağacı kullanarak çözümleyiniz. N'in, 3'ün pozitif bir üssü olduğunu varsayınız.

$$T(n) = 3T(n/3) + n$$

$$T(n) = 9T(n/3) + 1$$

$$T(n) = 9T(n/3) + 1$$

$$h = \log_3 n$$

$$T(n) = 9T(n/3) + 1$$

$$T(n/3) \quad T(n/3) \quad T(n/3) \quad T(n/3) \quad T(n/3) \quad T(n/3) \quad T(n/3) \quad T(n/3) \quad T(n/3)$$

$$T(n) = \sum_{i=0}^{\log_3 n} 9^i = \frac{9^{(\log_3 n)+1} - 1}{9 - 1} = \frac{9 * 9^{\log_3 n} - 1}{8} = \frac{9n^{\log_3 9} - 1}{8} = \frac{9 * n^2 - 1}{8} = O(n^2)$$

```

oku(N)
A=100;
say1=0;
say2=0;
say3=0;
for i=1:N
    s1=round(rand()*A);
    s2=round(rand()*A);
    if (s1*s1+s2*s2>A*A) && (s1>s2)
        say1=say1+1;
    end
    if (s1*s1+s2*s2<A*A) && (s1<s2)
        say3=say3+1;
    end
    if (s1<s2)
        say2=say2+1;
    end
end
yaz((say1+say3)/say2)

```

Yandaki sözde kod çalıştırıldığında $(say1+say3)/say2$ değeri N 'in büyük değerleri için yaklaşık olarak kaçta yakınsar?
 $rand()$: (0,1) açık aralığında ondalıklı rasgele bir sayı üretir.
 $round(x)$: x 'i aşağı ya da yukarı (hangisine yakınsa) yuvarlar.
 Çözümünüzü bir şekil üzerinde gösteriniz.

İpucu: $S1$ ve $S2$ 'yi bir noktanın 2 koordinatı şeklinde düşünebilirsiniz.

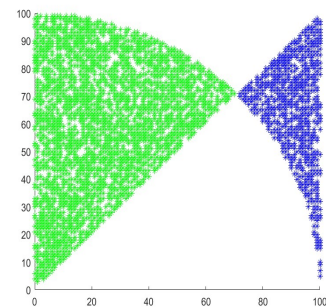
```

oku(N)
A=100;
say1=0;
say2=0;
say3=0;
for i=1:N
    s1=round(rand()*A);
    s2=round(rand()*A);
    if (s1*s1+s2*s2>A*A) && (s1>s2)
        say1=say1+1;
    end
    if (s1*s1+s2*s2<A*A) && (s1<s2)
        say3=say3+1;
    end
    if (s1<s2)
        say2=say2+1;
    end
end
yaz((say1+say3)/say2)

```

Yandaki sözde kod çalıştırıldığında $(say1+say3)/say2$ değeri N 'in büyük değerleri için yaklaşık olarak kaçta yakınsar? (20P)
 $rand()$: (0,1) açık aralığında ondalıklı rasgele bir sayı üretir.
 $round(x)$: x 'i aşağı ya da yukarı (hangisine yakınsa) yuvarlar.
 Çözümünüzü bir şekil üzerinde gösteriniz.

İpucu: $S1$ ve $S2$ 'yi bir noktanın 2 koordinatı şeklinde düşünebilirsiniz.



Say1= mavi bölgenin alanı

Say3= yeşil bölgenin alanı

Say2= 100*100'luk karenin yarısının alanı

Buna göre $(Say1+say3)/say2$ 'in değeri N 'in büyük değerleri için 1'e yakınsayacaktır.

```

oku(N)
T=0;
D=1;
while (D<N)
    D=D*3;
    T=T+1;
end
R=1;
for i=1:T
    for j=1:D
        R=R*3;
    end
    R=R*2;
end

```

Yandaki sözde kod çalışmasını bitirdiğinde D, T ve R 'nin değerlerini N'e bağlı olarak bulunuz. N, 3'ün pozitif bir üssü olarak girilmektedir. (Ör: N=243, N=27 vb.)

```

oku(N)
T=0;
D=1;
while (D<N)
    D=D*3;
    T=T+1;
end
R=1;
for i=1:T
    for j=1:D
        R=R*3;
    end
    R=R*2;
end

```

Yandaki sözde kod çalışmasını bitirdiğinde D, T ve R 'nin değerlerini N'e bağlı olarak bulunuz. N, 3'ün pozitif bir üssü olarak girilmektedir. (Ör: N=243, N=27 vb.)

$$\begin{aligned}
 D &= N \\
 T &= \log_3 N \\
 R &= 2^{\log_3 N} * 3^{N * \log_3 N} = 2^{\log_3 N} * N^N
 \end{aligned}$$