

Öğrencinin Adı Soyadı:	Öğrenci No:	İmza:
Dersin Adı: Yapısal Programlamaya Giriş	Tarih/Saat: 30.04.2019 09:00	Sınav süresi: 90
Sınav Türü:	Vize 2	
Unvan Ad-Soyad: Doç. Dr. M. Fatih AMASYALI, Dr. Öğr. Üyesi. H. İrem TÜRKMEN (Ders Yürütücüsü)		

1. 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 (20p).

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

```
}
```

2. $N \times N$ 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 (Pow) 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. $Pow < N-1$ olduğuna göre, kullanıcıdan alınan oyun alanındaki bombayı patlatarak oyun alanının son halini ekrana yazdıran programın akış diyagramını çizin (30p).

Örn:

Oyun Alanının
İlk Hali:

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 Son
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

```
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;
}
```

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

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

- a) Çalışma zamanı $T(n) = 3 \cdot T(n-3) + 1$ şeklinde verilen bir özyinelemeli fonksiyon için çalışma zamanını özyineleme ağacı kullanarak çözümleyiniz. Ağaç yüksekliği (5P), çalışma zamanı (10P)

$$h = n/3$$

$$T(n) = (3^{(n/3)+1} - 1) / 2$$

- b) $T(n) = T(n/a) + b$ formundaki bir özyineleme fonksiyonunun çalışma zamanının $\log_5 n$ (logaritma 5 tabanında n) olması için a (5P) ve b (10P) yi bulunuz.

$$a = 5$$

$$b = 1$$

4. “ $A(n-1)=1$ ise sonlandır, $A(n-1)$ çift sayı ise $A(n)=A(n-1)/2$, $A(n-1)$ tek sayı ise $A(n)=3 \cdot A(n-1)+1$ ” kuralıyla bir dizi üretimi yapılmaktadır. Örneğin dizi 13 ile başlarsa [$A(1)=13$] dizi 13 40 20 10 5 16 8 4 2 1 şeklinde olacak ve uzunluğu 10 olacaktır. Örneğin dizi 32 ile başlarsa [$A(1)=32$] dizi 32 16 8 4 2 1 şeklinde olacak ve uzunluğu 6 olacaktır.

“1-1000 arası sayılardan hangisi bu dizinin başlangıç değeri olursa en uzun dizi elde edilir?” sorusunu cevaplayan algoritmayı çiziniz (20p).

```
max_uz=1
```

```
max_uz_bas=0
```

```
for i=1:1000
```

```
    dizi_uz=dizi_olustur(i)
```

```
    if dizi_uz>max_uz
```

```
        max_uz=dizi_uz
```

```
        max_uz_bas=i
```

```
    end
```

```
end
```

```
dizi_uz fonksiyon dizi_olustur(bas)
```

```
    eleman=bas
```

```
    dizi_uz=0
```

```
    while eleman<>1
```

```
        dizi_uz=dizi_uz+1
```

```
        if mod(eleman,2)==0
```

```
            eleman=eleman/2
```

```
        else
```

```
            eleman=3*eleman+1
```

```
        end
```

```
    end
```

```
    return dizi_uz
```

