VERİ YAPILARI-VİZE SINAVI (I. Ve II. Öğretim)

S.1. a) A / B ^ C + D * E – A * C ifadesini postfix ifadeye dönüştürünüz. Yönteminizi açıklayınız.
b)6 2 3 + - 3 8 2 / + * 2 ^3 + ifadesinin sonucunu hesaplayınız. Yönteminizi açıklayınız.

Sembol	Operand1	Operand 2	Değer	Yığıt

S.2. 10 yataklı bir hastanede hasta-yatak takibi için bağlı liste kullanarak bir işletim gerçekleyiniz. Örneklerle yönteminizi açıklayınız.

S.3. Tek boyutlu ve iki boyutlu bir dizi için adres polinomu yazınız. Dizi ile bağlı listeyi karşılaştırınız.

S.4. Çift yönlü sıralı doğrusal bir bağlı liste liste ortasına eleman eklemek ve silmek için bir formül (algoritma) geliştiriniz. Birer örnekle algoritmanızı çalıştırınız.

S.5. Aşağıdaki algoritmanın analizini ayrıntılı olarak yaparak algoritma karmaşıklığını hesaplayınız.

```
for(i=1;i<=n;i++)
{ for(j=1;j<n;j++)
{ if(A[j+1]<A[j])
{ b=A[j];
A[j]=A[j+1];
A[j+1]=b;}
```

VERİ YAPILARI-VİZE SINAVI (I. Ve II. Öğretim)

1) "Dugum" ve "Liste" sınıfları aşağıda verilmiş olan ***tek yönlü bağlı liste*** yapısında sayısal veriler ***küçükt büyüğe*** olacak şekilde tutulmaktadır. Buna göre bağlı liste yapısına yeni bir eleman ekleme işini gerçekleştirecek ***"ekle"*** fonksiyonunu yazınız. (c veya c++)

Not: "ekle" fonksiyonunun parametre olarak ne alacağı size bağlıdır. İstenirse parametresiz de yazılabilir. (ekle(), ekle (int a), ekle (Dugum yeni))

```
class Dugum
{ public: int sayi;
        Dugum *sonraki;
        Dugum();
        Dugum(int a){ sayi=a; }
}; // Dugum sınıfı sonu
```

```
class Liste
{ public: Dugum *bas;
        Dugum *son;
        Liste(){bas=0;son=0;}
//ekle fonksiyonu
}; // Liste sınıfı sonu
```

2) Bir telefon defterindeki numaralar ***dairesel kuyruk*** veri yapısı kullanılarak tutulacaktır. Yeni bir telefon numarasını telefon defterine ekleme işlemini gerçekleştiren ***"ekle"*** fonksiyonunu yazınız. (c veya c++)

Not: "ekle" fonksiyonunun parametre olarak ne alacağı size bağlıdır. İstenirse parametresiz de yazılab (ekle(), ekle (int telefonNo)). Telefon defterinde en fazla 10 numara olacağını varsayabilirsiniz. Kullanacağınız "global" değişkenleri fonksiyonu yazmadan önce tanımlamayı unutmayınız(c veya c+

3) Bir strançoyununda yapılan hamleler ***stack(yığın)*** veri yapısı kullanılarak tutulmak istenmekte Yapılan her hamle sonrası eğer stack yapısı boş ise hamle stack'e eklenecektir(push fonksiyonu). Eğer yapılan son hamle geri alınmak istenirse de stackten bu veri çıkartılcaktır(pop fonksiyonu). Ayrıca oyunun bitiminde stack'te bulunan tüm hamleler teker teker ekrana yazdırılacaktır(yazdır fonksiyonu). Bu özelliklere göre ekleme işini gerçekleştiren ***push***, çıkarma işlemini gerçekleştiren ***pop*** ve tüm veriyi ekrana yazdıran ***yazdır*** fonksiyonlarını C veya C++ programlama dillerini kullanarak gerçekleştiriniz.

Not: fonksiyonların parametre alıp almayacağı veya ne parametre alacağı size bağlıdır. İstenilen sadec tane fonksiyonun kodlanmasının gerçekleşmesidir ***programın tamamı istenmemektedir.*** Eğer kullanmanız gereken ***global*** değişkenler varsa bunları tanımlamayı unutmayınız