GİRESUN ÜNİVERSİTESİ MÜHENDİSLİK FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ
VERİ YAPILARI DERSİ VİZE SINAVI

Adı – Soyadı – Numarası:

Soru 1: Parametre olarak verilen dizinin, ters çevrilmiş halini geri döndüren metodu yazınız. (15 puan)

```
public int[] diziTersCevir(int[] dizi) {
	int[] tersDizi = new int[dizi.length];
	for (int i = 0; i < dizi.length; i++) {
		tersDizi[i] = dizi[dizi.length - 1 - i];
	}
	return tersDizi;
}
```

```
public int[] diziTersCevir(int[] dizi) {
	int uzunluk = dizi.length;
	for (int i = 0; i < uzunluk / 2; i++) {
		int gecici = dizi[i];
		dizi[i] = dizi[uzunluk - 1 - i];
		dizi[uzunluk - 1 - i] = gecici;
	}
	return dizi;
}
```

```
public int[] diziTersCevir(int[] dizi) {
	Stack<Integer> stack = new Stack<>();
	for (int i = 0; i < dizi.length; i++) {
		stack.push(dizi[i]);
	}
	for (int i = 0; i < dizi.length; i++) {
		dizi[i] = stack.pop();
	}
	return dizi;
}
```

Soru 2: Parametre olarak verilen iki dizi içerisinde eşleşen elemanları yazdıran metodu yazınız. (15 puan)

```
public void eslesenleriBulVeYazdir(int[] dizi1, int[] dizi2) {
	for ( int eleman1 : dizi1 ) {
		for ( int eleman2 : dizi2 ) {
			if (eleman1 == eleman2) {
				System.out.print(eleman1 + " ");
				break;
			}
		}
	}
}
```

Soru 3: Çift yönlü bağlı listenin sonuna eleman ekleyen metodu tamamlayınız. (20 puan)

Başarılar.. Dr. Öğr. Üyesi Sercan KÜLCÜ 20.11.2023

```
public void sonaEkle(int veri) {
	CiftYonluDugum yeniDugum = new CiftYonluDugum(veri);
	if (bas == null) {
	// Liste boşsa, yeni düğüm hem başı hem de sonu temsil eder
		bas = yeniDugum;
		son = yeniDugum;
	} else {
	// Liste boş değilse, yeni düğümü listenin sonuna ekler
		yeniDugum.onceki = son;
		son.sonraki = yeniDugum;
		son = yeniDugum;
	}
}
```

Soru 4: Aşağıdaki metodu açıklayınız ve yaptığı işi tanımlayınız. (20 puan)

```
private int fonksiyon() {
	TekYonluDugum a = bas;
	TekYonluDugum b = bas;
	while (b != null && b.sonraki != null) {
		a = a.sonraki;
		b = b.sonraki.sonraki;
	}
	return a.veri;
}
```

Bu metodun içerisinde bir bağlı liste (linked list) üzerinde gezinme işlemi gerçekleştirilmektedir. Metodun amacı, bağlı listenin ortasındaki elemanı bulmaktır. Bu metodun çalışma prensibi, iki referansın biri diğerinden iki kat daha hızlı bir şekilde bağlı listede ilerlemesidir. Bu sayede döngü sona erdiğinde yavaş ilerleyen referansın bulunduğu konum, bağlı listenin ortasına denk gelmiş olur.

Soru 5: Aşağıdaki program çalıştırıldığında çıktısı ne olur? (15 puan)

```
public static void main(String args[]) {
	Stack<String> stack = new Stack<>();
	stack.push("A");
	stack.push("C");
	stack.push("D");
	stack.push("B");
	System.out.print(stack.pop() + " ");
	System.out.print(stack.peek() + " ");
	System.out.print(stack.remove("D") + " ");
```

Başarılar.. Dr. Öğr. Üyesi Sercan KÜLCÜ 20.11.2023

```
        System.out.print(stack.pop() + " ");
        System.out.print(stack.pop() + " ");
        System.out.print(stack.pop() + " ");
}
```

B D true C A Exception

Soru 6: Aşağıdaki program çalıştırıldığında çıktısı ne olur? (15 puan)

```
public static void main(String args[]) {
        Stack<String> s = new Stack<>();
        Queue<String> q = new LinkedList<>();
        q.offer("1");
        q.offer("3");
        q.offer("5");
        q.offer("7");
        while(!q.isEmpty())
                s.push(q.poll());
        while(!s.isEmpty())
                q.offer(s.pop());
        System.out.println("s: " + s + " --- q: " + q);
}
```

s: [] --- q: [7, 5, 3, 1]