



CEVAPLAR

1. Lütfen boş bırakılan yerlere gelmesi gereken ifadeleri cevap kâğıdınıza geçirin.

- a) sonsuza doğru / infinite
- b) soyut veri türü / abstract data type
- c) sabit / fixed / O(1)
- d) sabit / fixed / O(1)
- e) dinamik / dynamic
- f) kafa / baş / head
- g) doğrusaldır / O(n)
- h) ekstra bellek / işaretçi
- i) son giren ilk çıkar / last in first out (LIFO) / ilk giren son çıkar / First in Last out (FILO)
- j) peek / top

2. "Contains duplicate" probleminin çözümü

```
1 public bool ContainsDuplicate(T[] nums)
2 {
3     HashSet<T> set = new HashSet<T>();
4
5     foreach (var x in nums)
6     {
7         if (set.Contains(x))
8             return true;
9         set.Add(x);
10    }
11    return false;
12 }
```

3. "Reverse Linked List" problemi çözümü:

```
1 public SinglyLinkedList<T> ReverseList(SinglyLinkedListNode<T> head)
2 {
3     if (head is null)
4         throw new ArgumentNullException("head");
5
6     var reverseList = new SinglyLinkedList<T>();
7     var curr = head;
8
9     while (curr is not null)
10    {
11        reverseList.AddBefore(reverseList.Head, curr.Value);
12        curr = curr.Next;
13    }
14    return reverseList;
15 }
```

4. Çift yönlü bağlı listede (**DoublyLinkedList<T>**) liste sonundan liste başına doğru gezinme sağlayan **DbLinkedListReverseEnumerator** sınıfında **MoveNext()** işlevinin çözümü:

```
1 public bool MoveNext()
2 {
3     if (CurrentNode == null)
4         return false;
5
6     CurrentNode = CurrentNode.Prev;
7     return CurrentNode != null;
8 }
```

5. Dizinin bağlı listeye çevrilmesine ilişkin çözüm.

```
1 public SinglyLinkedList<T> ToSinglyLinkedList()
2 {
3     var linkedList = new SinglyLinkedList<T>();
4
5     foreach (var item in _innerArray)
6     {
7         linkedList.AddLast(item);
8     }
9
10    return linkedList;
11 }
```