

Write a recursive method that takes a `String` parameter and prints out the characters of the string in reverse order.

```
public static void reverse(String s){  
  
}
```

Write a recursive method that takes an integer parameter (n) and return the following summation:

$$\sum_{i=1}^n \lim_{i \rightarrow n}$$

```
public int sum(int n) {  
  
}
```

Write a recursive method that take an integer parameter (n) and counts the number of digits of n

```
public int countDigits(int n) {  
  
}
```

Write a recursive method that takes an integer parameter (n) and prints it in reverse

```
public void printReverse(int n)  
{  
  
  
}
```

Write the code fragment to insert a new node that the reference variable `newNode` references before the node referenced by the reference variable `curr` in a doubly linked list.

Write the code fragment to insert a new node that the reference variable `newNode` references after the node referenced by the reference variable `curr` in a doubly linked list.

Given `MyLinkedList<E>` class.

```
public class MyLinkedList<E> implements MyList<E> {  
    private Node<E> head, tail;  
    private int size = 0; // Number of elements in the list  
  
}
```

Implement the following method:

```
/** Remove the element at the specified position in this list  
 *   Shift any subsequent elements to the left.  
 *   Return the element that was removed from the list. */  
public E remove(int index);
```

```
/** Remove the first occurrence of given element from this list  
 *   Shift any subsequent elements to the left.  
 *   Return the element that was removed from the list.  
 *   Return null if element not found */  
public E remove(E e);
```

- What is the order of each of the following functions?

- (a) $(n^2 + 1)^2/n$
(b) $(n^2 + \log^2 n)^2 / n$
(c) $n^3 + 100n^2 + n$
(d) $2^n + 100n^2 + 45n$
(e) $n2^n + n^22^n$

- Count the number of iterations in the following loops.

(a)

```
int count = 1;  
while (count < 30) {  
    count = count * 2;  
}
```

(b)

```
int count = 15;  
while (count < 30) {  
    count = count * 3;  
}
```

(c)

```
int count = 1;  
while (count < n) {  
    count = count * 2;  
}
```

(d)

```
int count = 15;  
while (count < n) {  
    count = count * 3;  
}
```

- Put the following growth functions in order: $5n^3/4032$, $44\log n$, $10n\log n$, 500 , $2n^2$, $2^n/45$, $3n$
- Describe an algorithm for removing duplicates from an array. Analyze the complexity of the algorithm.

