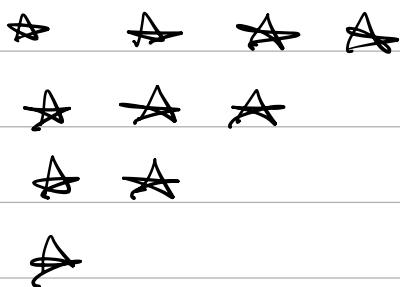


# Recursion Pattern Questions

Q).



$\text{fun}(r, c)$

$\text{fun}(h, 0)$

$\downarrow$

$\text{fun}(h, 1)$

$\downarrow$

$f(e_1, 2)$

$f(h, 3)$

$\downarrow$

$f(h, h) \times \underbrace{L}_{\text{!}} \leftarrow ! < h.$

$\downarrow$

newline

$f(3, 0)$

$\downarrow$

$f(3, 1)$

$\downarrow$

$\vdots$

$\vdots$

$\{\}$

Logic:-

fun( row, col )

if ( row == col )  
 fun( row - 1, 0 );

else

if ( col < row )  
 sout( "\*" )  
 fun( row, col + 1 );

else  
 return;

}

Code:

```

3 ► public class Triangle {
4 ►   public static void main(String[] args) {
5     triangle(r: 4, c: 0);
6   }
7
8   static void triangle(int r, int c) {
9     if (r == 0) {
10       return;
11     }
12     if (c < r) {
13       System.out.print("*");
14     triangle(r, c: c+1);
15   } else {
16     System.out.println();
17     triangle(r: r-1, c: 0);
18   }
19 }
```

Q).

\*

\* \*

\* \* \*

\* \* \* \*

Soln

Logic:-

fun(row, col) {

if (row == 0) {  
 return;

}

else

if (row > col) {

fun (row, col + 1);

cout(<<\*);

}

else {

fun (row - 1, 0);

cout(<<#);

}

}

NOTE:-

Only when function call  
finishes execution, cout will execute

## Code:-

```
        triangle2(r: 4, c: 0);  
    }  
  
    static void triangle2(int r, int c) {  
        if (r == 0) {  
            return;  
        }  
        if (c < r) {  
            triangle2(r, c: c+1);  
            System.out.print("*");  
        } else {  
            triangle2(r: r-1, c: 0);  
            System.out.println();  
        }  
    }  
}
```

## Bubble sort in recursion

### Logic:-

```
void bubble (int arr[], int n, int c)
```

{

```
if (n == 0)
```

```
return;
```

```
if (c < n) {
```

```
if (arr[c] > arr[c+1]) {
```

```
// swap }
```

```
bubble (arr, n, c+1);
```

}

else {  
    bubble (arr, r-1, c);  
}  
}.

Here you are passing last element index  
as r and c as 0.  
→ start index.

Code:-

```
static void bubble(int[] arr, int r, int c) {  
    if (r == 0) {  
        return;  
    }  
    if (c < r) {  
  
        if (arr[c] > arr[c+1]) {  
            // swap  
            int temp = arr[c];  
            arr[c] = arr[c+1];  
            arr[c+1] = temp;  
        }  
        I  
        bubble(arr, r, c:c+1);  
    } else {  
        bubble(arr, r:r-1, c:0);  
    }  
}
```

last index.

→ start index.

# Selection sort using recursion

Logic:- Same Logic as before,  
just pass a maximum index  
as argument.

Code:-

```
static void selection(int[] arr, int r, int c, int max) {  
    if (r == 0) {  
        return;  
    }  
    if (c < r) {  
        if (arr[c] > arr[max]) {  
            selection(arr, r, c+1, c);  
        } else {  
            selection(arr, r, c+1, max);  
        }  
    } else {  
        int temp = arr[max];  
        arr[r-1] = arr[max];  
        arr[max] = temp;  
        selection(arr, r-1, 0);  
    }  
}
```

selection (arr, r-1, 0, 0)  
startindex      lastindex      mark index