

Workshop #7: Recursion

GitHub: abhijit-baruah

Group: Jen Rowe, Charlotte Ding and Abhijit Baruah

Take this recursive function:

```
1 def fun(n):
2     if n == 0:
3         return []
4
5     return fun(n//2) + [n%2]
6
7 fun(25)
```

Ans #a:

Line number for the base case is 2

Ans #b:

The line number, and specific recursive call in fun(n) are: 5 and fun(n//2) respectively.

Ans #c:

| RecursionLevel | n | fun(n) | is n==0? | n//2 | n%2 |
|----------------|----|---------|----------|------|-----|
| 0 | 25 | fun(25) | FALSE | 12 | 1 |
| 1 | 12 | fun(12) | FALSE | 6 | 0 |
| 2 | 6 | fun(6) | FALSE | 3 | 0 |
| 3 | 3 | fun(3) | FALSE | 1 | 1 |
| 4 | 1 | fun(1) | FALSE | 0 | 1 |
| 5 | 0 | fun(0) | TRUE | | |

Ans #d

Final return code is: [1 1 0 0 1]