

**PROBLEM 4-1** (2/2 points)

Consider the following Python procedure. Specify its order of growth.

```
def modten(n):  
    return n%10
```

You have used 1 of 1 submissions

PROBLEM 4-2 (2/2 points)

Consider the following Python procedure. Specify its order of growth.



```
def multlist(m, n):  
    '''  
    m is the multiplication factor  
    n is a list.  
    '''  
    result = []  
    for i in range(len(n)):  
        result.append(m*n[i])  
    return result
```

You have used 1 of 1 submissions

PROBLEM 4-3 (2/2 points)

Consider the following Python procedure. Specify its order of growth.

```
def foo(n):  
    if n <= 1:  
        return 1  
    return foo(n/2) + 1
```



 

You have used 1 of 1 submissions

PROBLEM 4-4 (2/2 points)

Consider the following Python procedure. Specify its order of growth.

```
def recur(n):  
    if n <= 0:  
        return 1  
    else:  
        return n*recur(n-1)
```

You have used 1 of 1 submissions

PROBLEM 4-5 (2/2 points)

Consider the following Python procedure. Specify its order of growth.

```
def baz(n):  
    for i in range(n):  
        for j in range(n):  
            print i,j
```



You have used 1 of 1 submissions

Show related resources



© All Rights Reserved



[About](#) [Blog](#) [News](#) [FAQs](#) [Contact](#) [Jobs](#) [Donate](#) [Sitemap](#)

[Terms of Service & Honor Code](#) [Privacy Policy](#) [Accessibility Policy](#)

© edX Inc. All rights reserved except where noted. EdX, Open edX and the edX and Open EdX logos are registered trademarks or trademarks of edX Inc.

POWERED BY
OPENedX

