

\* This code is my own work. It was written without consulting code written by other students or code from online resources. Felipe Cardozo\*/

4 a)  $m=1$   
 $n=4$   
 else  $\rightarrow$  print ("continue") ~~if  $n \neq 4$~~  ~~variables = m, n~~  
 $\rightarrow$  return

$\rightarrow n: m \cdot \text{recurse}$   
 $m=1$   
 $n=1$   
 $\rightarrow$  prod(1,1)  
 if  $\rightarrow$  print ("done")  
 $\rightarrow n: 1$

$\rightarrow$  prod(1,2)  
 $m=1$  else  $\rightarrow$  print ("continue")  
 $n=2$

$\rightarrow$  prod(1,3)  
 else  $\rightarrow$  print ("continue")  
 $n=3$

$\rightarrow$  prod(1,4)  
 else  $\rightarrow$  print ("continue")  
 $n=4$

$\rightarrow$  Return  $4 \cdot 3 \cdot 2 \cdot 1 = 24$

b) continue 14  
 continue 13  
 done 12  
 24

c) The method checks if  $m=n$ . When  $m \neq n$ , else is executed, ~~when  $m=n$~~  and the program continues, when  $m=n$ , it stop and calculates the product of  $n$  to 1 times  $m$ .

2b)  $m=1$   
 $n=1$   
 else  $\rightarrow$  print ("continue")  $\rightarrow$  continue the recursion  
 $\rightarrow$  ack(0, ack(1,0))  
 else if  $\rightarrow$  print ("continue")  $\rightarrow$  continue the recursion  
 $\rightarrow$  ack(0,1)  
 if  $\rightarrow$  print ("maybe done")  
 $\rightarrow n: n+1=2$   
 this is inside a recursion not a actual return (out put)  
 $\rightarrow$  ack(0,1)  
 if  $\rightarrow$  print ("maybe done")  
 $\rightarrow n: n+1=3$   
 $\rightarrow$  Return 3

3b)  $s = "abca"$   
 $c = 'a'$

if  $\rightarrow s.charAt(0) == 'a'$ , remove  $s.charAt(0)$   
 True

$\rightarrow removeLetter("abca", 'a')$

if  $\rightarrow s.charAt(0) == 'a'$ , remove  $s.charAt(0)$   
 true

$\rightarrow removeLetter("bca", 'a')$

if  $\rightarrow s.charAt(0) == 'a'$ , keep  $s.charAt(0)$   
 False

$\rightarrow removeLetter("ca", 'a')$

if  $\rightarrow s.charAt(0) == 'a'$ , keep  $s.charAt(0)$   
 False

$\rightarrow removeLetter("a", 'a')$

$\rightarrow s.charAt(0) == 'a'$ , remove  $s.charAt(0)$

$\rightarrow removeLetter("", 'a')$

if  $\rightarrow return ""$

$\rightarrow Return("bc")$

4. b)  $s = "abc"$

$\rightarrow call/continue recursion$

$s = "bc"$

$\rightarrow s.length() < 2 \parallel s.charAt(0) > s.charAt(1)$   
 False

$\rightarrow call/continue recursion$

$s = "c"$

$\rightarrow s.length() < 2 \parallel s.charAt(0) > s.charAt(1)$   
 False

$\rightarrow s = ""$

$\rightarrow s.length() < 2 \parallel s.charAt(0) > s.charAt(1)$   
 True

True

$\rightarrow Return [True]$