

Given the following code, draw the recursive trace in memory and indicate what the final result will be for the function call below:

```
1 def fibonacci( int_x ):
2
3     print( 'Call:', int_x )
4
5     # Base Case
6     if int_x <= 1:
7         return 1
8
9     # Recursive case
10    return fibonacci(int_x - 1) + fibonacci(int_x - 2)
11
12
13 fibonacci(5)
```

Given the following code, draw the recursive trace in memory and indicate what the final result will be for the function call below:

```
1 def prob_3( int_val ):
2
3     if int_val <= 1:
4         return 2
5
6     if int_val % 2 == 1:
7         return int_val + prob_3( int_val - 2 )
8
9     else:
10        return int_val + prob_3( int_val - 1 )
11
12 prob_3( 8 )
```

Given the following code, draw the recursive trace in memory and indicate what the final result will be for the function call below:

```
1 def prob_4( x ):  
2  
3     if x <= 3:  
4         return 2  
5  
6     return prob_4( x - 2 ) + prob_4( x - 3 )  
7  
8 prob_4(6)
```

Given the following code, draw the recursive trace in memory and indicate what the final result will be for the function call below:

```
1 def foo(x):
2
3     if x <= 0:
4         return 5
5
6     if x % 2 == 0:
7         return x + bar(x-1)
8
9     elif x % 2 == 1:
10        return x + foo(x-3)
11
12
13 def bar(x):
14
15     if x <= 2:
16         return 4
17
18     if x % 2 == 0:
19         return x + foo(x-3)
20
21     elif x % 2 == 1:
22         return x + bar(x-1)
23
24
25 foo(12)
```