
Code tracing

Use the following piece of code to answer question 2 and 3

```
def bits(n):  
    s = ''  
  
    while n != 0:  
        if n % 2 == 0:  
            s = '0' + s  
        else:  
            s = '1' + s  
  
        n = n/2  
  
    return s
```

2. What is the return value of `bits(8)`? (10 points)

3. What is the return value of `bits(11)`? (10 points)

```
def f(n):
    if n == 0:
        return 0
    elif n % 2 == 0:
        return f(n) + n
    else:
        return f(n - 1) - n
```

4. There is a problem with the above function. What is it? Give one way to fix this problem (you may simply cross out the broken line and then write the corrected version of that line next to it or in the space below.) (10 points)
5. Show the stack diagram *of your fixed version* for the call $\text{fib}(5)$. (10 points)

Code Writing

```
def saying_one_____:
    x = -1
    while _____:
        x = random.randint(1, 4)

    part = ''
    if x == 1:
        part = 'liver'
    elif x == 2:
        part = 'spleen'
    elif x == 3:
        part = 'duodenum'
    elif x == 4:
        part = 'femur'
    else:
        part = 'invalid'

    return prefix + ' my ' + part + "'s acting up"
```

6. Fill in the blanks in the above piece of code to make the assertion

`assertEqual(saying_one('Aww jeez'), "Aww jeez my duodenum's acting up")` hold.
(10 points)

-
1. Implement a function `primes_between` which takes two integers x and y (such that $x < y$) and *prints to the screen* all prime numbers between x and y . (15 points)

-
2. Implement a function `digits` which takes an integer n and returns the number of digits in n . Your function should be able to handle negative integers. **Hint:** Consider implementing this one recursively. (15 points)