

1) (5 pts) DSN (Recursive Coding)

Write a **recursive** function that returns the number of bits set to 1 in the binary representation of its input parameter, n . (Note: In order to receive full credit, your function's run time must be $O(b)$, where b is the total number of bits in n . Since this isn't the bitwise operator question, you don't HAVE to use bitwise operators for full credit, but that's probably the most natural route to the solution.)

```
int numBitsOn(int n) {  
  
    // Base case - 2 pts  
    if (n == 0)  
        return 0;  
  
    // 1 pt extract lsb, 1 pt add, 1 pt recursive call  
    return (n&1) + numBitsOn(n>>1);  
}
```

Note: Bitwise operators aren't needed, so $n\%2 == 1$ for the bit check and $n//2$ for the input to the recursive call also get full credit. Also, one can make a second if statement and get equivalent behavior, so just watch out for correct solutions that are expressed differently and make sure to award all of these full credit. Map the points as shown above for each subtask.

Correct non-recursive solution: 3/5

Non-recursive solution with minor bug: 2/5