631541
How many number will be printed if n = 5?

```
public static void cSeries(int n) {
      System.out.print(n + " ");
      if (n == 1) return;
      if (n % 2 == 0) cSeries(n / 2);
      else cSeries(3*n + 1);
}
```

a. 3

b. 4

c. 5

*d. 6

e.
f. "
g. "
h. "
i. "

General Feedback:

5

632300
For Club Scripting, to be a member a person must be aged 15 or older, but less than
50. What values would you use to boundary test a script designed to check if a
person was a suitable age to join?

a. 14, 15, 50, 51

b. 15, 16, 50, 51

c.
15, 16, 49, 50

*d.
14, 15, 49, 50

e. 13, 15, 49, 52

f. "
g. "
h. "
i. "
j. "

General Feedback:

The boundary values are those at the boundary and those respectively one before and one past the boundary

634252
Read the following method skeleton and choose the best expression to fill in the blank on **line 5** so that the method will behave correctly:

```
/**
* Takes a string reference and counts the number of times
* the character 'A' or 'a' appears in the string object.
* @param aString   String reference to object containing chars.
* @precondition    aString is not null (you may assume this is true).
* @return          The number of times 'A' or 'a' appears in the string.
*/
public static int countAs(String aString)  // line 1
{
    int counter = __________;              // line 2
    int totalA = 0;                        // line 3
    while (counter < __________)           // line 4
    {
        if ( __________.equals("A") )      // line 5
        {
            totalA = totalA + __________;  // line 6
        }
        counter++;                         // line 7
    }
    return __________;                     // line 8
}
```

a. aString.charAt(counter)
b. aString.substring(counter)
c. aString.charAt(counter).toUpperCase()
d. aString.substring(counter).toUpperCase()
*e. aString.substring(counter, counter + 1).toUpperCase()
f. "
g. "
h. "
i. "
j. "

General Feedback:

While both `charAt()` can be used to examine a single `char` in a string, `char` is a primitive type without any methods. Since the condition on Line 5 uses the `equals()` method to compare against another String value, the expression used to fill in the blank must produce a String object, so the `substring()` method is a better fit. The `substring()` method takes two parameters--a starting index and an ending index one past the end of the substring--and since the method counts both upper case and lower case A's, the result should be converted to upper case form before testing for equality against `"A"`. Thus, the best answer is `aString.substring(counter, counter + 1).toUpperCase()`.