

634254

Read the following method skeleton and choose the best expression to fill in the blank on **line 8** 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. counter
- b. true
- c. false
- *d. totalA
- e. aString
- f. "
- g. "
- h. "
- i. "
- j. "

General Feedback:

The return type of the method is `int`, so an integer return value must be provided. Since `counter` is used as a loop index, it will always end up being the total number of characters in the given string. The variable `totalA` is used as an accumulator that is incremented each time a letter A is found in the string, so it is the choice that will provide the correct return value for the method.

633247

Two algorithms accomplish the same task on a collection of N items. Algorithm A performs $\log_2 N$ operations. Algorithm B performs $\log_3 N$ operations. Under what conditions does algorithm A offer better performance?

- a. $N \leq 2$
- b. $N < \log_2 3$
- c. $N < \log_3 2$

d. $N < 8$

*e. For no N .

f. "

g. "

h. "

i. "

j. "

General Feedback:

For no possible collection size N is $\log_2 N < \log_3 N$.

633241

Finding the median value in a complete and balanced binary search tree is

*a. $O(1)$

b. $O(\log N)$

c. $O(N)$

d. $O(N^2)$

e. $O(N \log N)$

f. "

g. "

h. "

i. "

j. "

General Feedback:

The median is the element that has M elements less than it and M elements greater than it. This can only be said of the root node in a complete and balanced binary tree. The root is accessed in constant time.