Assertions and Invariants

A loop invariant is an assertion about the loop that is relevant to the purpose of the loop and that holds true before and after each iteration through the loop. This assertion is usually expressed as a relation between the variables involved in the loop. Loop invariants are used to reason about programs formally and to prove their correctness without tracing all the iterations through a loop. If you can establish that an assertion is true before the first iteration, and also prove that for any iteration if the assertion is true before that iteration it will remain true after that iteration, then your assertion is a loop invariant. If you are familiar with mathematical induction, you can see how it works here. If not, you can still answer questions about loop invariants without too much trouble.

Example:

Consider the following code segment:

```
int count = 0;
int n = 41;
int k = 2;

while (k <= n)
{
  if (isPrime(k))
    count++;
  k++;
}
```

Which of the following statements are loop invariants for the above code?

I. *k* is a prime.
II. 41 is a prime.
III. `count` is equal to the number of primes from 2 to *k*-1.

(A) I only
(B) II only
(C) III only
(D) I and II
(E) None of the three

Statement I is not an invariant because it varies: `k` may or may not be a prime as we iterate through the loop. More precisely, `k` is a prime before the first iteration (`k` = 2) and before the second iteration (`k` = 3), but not after it (`k` = 4). This eliminates A and D. Statement II is not an invariant for a different reason. Certainly 41 is a prime — so are 37, 43, 47, and an infinite number of other integers. Also, Washington, DC, is the capital of the United States. These facts, while true, do not help us reason about the purpose or correctness of the above code. But Statement III is a typical invariant: it links the values of the variables `count` and `k` and reflects the purpose of the loop, namely counting all the primes from 2 to `n`. The answer is C.

Source: *Be Prepared for the AP Computer Science Exam in Java* by Maria Litvin.