Introduction to Software Testing

Question 1. Consider a program P, and two test suites, X and Y for P. Test suite X covers set of *branches* Q in the program, while test suite Y covers set of branches R in the program. Suppose Q is a proper subset of R ($Q \subset R$). Which of the below statements are necessarily true?

A. Whenever a test in Y reaches a statement, some test in X also reaches that statement.
B. Whenever a test in X reaches a statement, some test in Y also reaches that statement.
C. Test suite Y has strictly higher path coverage than test suite X.
D. Test suite Y has strictly higher branch coverage than test suite X.

Question 2. Consider the Java function:

```
void copy(int[] src, int[] dst, int N) {
    for (int i = 0; i < N; i++)
        dst[i] = src[i];
}
```

Which of the below predicates is the function's weakest possible precondition that prevent any null-pointer or array-out-of-bounds exceptions from being thrown?

NOTE: The expression X => Y is read: "X implies Y". It is equivalent to (!X) OR Y.

A. $N \geq 0 \land$ src != null $\land$ dst != null $\land$ N < src.length $\land$ N < dst.length
B. $N \geq 0 \land$ src != null $\land$ dst != null $\land$ N < src.length $\land$ dst.length = src.length
C. $N > 0 \Rightarrow$ (src != null $\land$ dst != null $\land$ N < src.length $\land$ N < dst.length)
D. $N > 0 \Rightarrow$ (src != null $\land$ dst != null $\land$ N < src.length $\land$ dst.length = src.length)

Question 3. Consider a test suite consisting of three deterministic tests T1, T2, T3 for a correct program P. Since P is correct, it passes all the three tests. Suppose we have three mutants M1, M2, M3 of P such that: M1 fails T1 and T2; M2 fails none; and M3 fails T2 and T3.

Let M = P denote that M and P are equivalent. Likewise, let M != P denote that they are NOT equivalent.

a. Could it be possible that M1 = P? Justify your answer.

b. Mutation analysis will report M2 to the tester since it does not fail any test in the test suite. Which of the following actions are plausible for the tester to take given this information?

A. Determine if M2 == P, and if so, devise a new test case T4 on which M2 passes but P fails.
B. Determine if M2 != P, and if so, then ignore M2.
C. Determine if M2 != P, and if so, devise a new test case T4 on which P passes but M2 fails.
D. Determine if M2 == P, and if so, ignore M2.