1. What if we use $|1\rangle \otimes |0\rangle$ instead of $|0\rangle \otimes |1\rangle$ as the inputs to Deutsch’s algorithm? Does the algorithm still distinguish constant from balanced functions? Explain.

2. What if we use $|1\rangle \otimes |1\rangle$ instead of $|0\rangle \otimes |1\rangle$ as the inputs to Deutsch’s algorithm? Does the algorithm still distinguish constant from balanced functions? Explain.

3. Consider a function $f(x)$ that takes 128-bit binary strings as input and returns a single bit as output. There are a total of $2^{128}$ possible input strings $x$. $f$ is constant if $f(x)$ is the same value (0 or 1) for every $x$. $f$ is balanced if $f(x)$ gives 0 for exactly half of the inputs, and 1 for the other half. Suppose that we know for sure that $f$ is either constant or balanced.

   (a) Best-case scenario: If we evaluate $f$ on successive inputs in some fixed order (so that we never evaluate $f$ more than once on the same input), using a classical computer, what is the minimum number of evaluations needed to determine whether $f$ is balanced or constant? Explain.

   (b) Worst-case scenario: What is the maximum number of evaluations needed, using a classical computer? Explain.