

**Developer Interview Coding Exercise**

Write a Java program that takes as input:

* The name of a chess piece (King, Queen, Bishop, Knight, Rook, Pawn)
* A starting digit, 0 to 9.

As output, the program should count all the legal phone numbers, beginning with the starting digit, that can be generated by moving the piece around a standard phone pad.

A legal phone number has the following requirements:

* Must be 10 digits long
* Can contain only digits (no \* or #)

The phone pad layout is

1 2 3   
4 5 6   
7 8 9   
\* 0 #

A legal move is defined as any move that would normally be legal for the piece in chess,   
with the following additions:

* Staying in place is a legal move for all pieces
* When a Pawn reaches the top row, it becomes a Queen. If the initial starting digit is on the top row, then the pawn must stay in place for its first move before becoming a queen.

For the case where the piece is a Pawn, starting on one of the bottom two rows (i.e. any of the digits 7, 8, 9 and 0), the Pawn may move either one or two spaces forward on its first move. If the pawn stays in place for its first move it may only move one space forwards thereafter.

Guidance notes:

* Return your source code at least 24 hours prior to your interview.
* There is no need to return a list of the actual numbers generated; just count them.
* Your program should return the answer ‘124768940’ when asked to count the number of moves for a Queen starting on position 5.
* Do not spend more than 4 hours on this exercise. Incomplete solutions will not be penalised, although the program should execute successfully.
* Your solution will be discussed in detail during the interview.