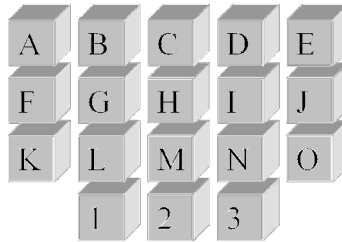


Knight Moves Problem

Pictured here is a keypad:



We want to find all sequences of length n that can be keyed into the keypad in the following manner:

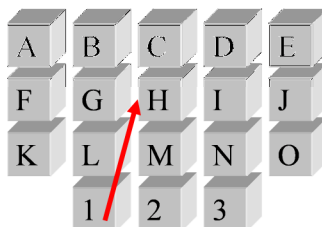
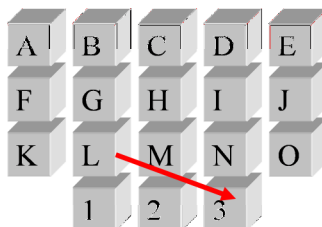
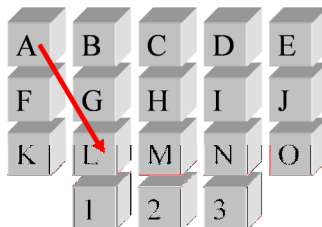
- The initial keypress can be any of the keys.
- Each subsequent keypress must be a *knight move* from the previous keypress.
- There can be at most 2 vowels in the sequence.
- We will attempt to run your solution for $n = 10, 16$ and 32 .

A knight move is made in one of the following ways:

- Move two steps horizontally and one step vertically.
- Move two steps vertically and one step horizontally.

There is no wrapping allowed on a knight move.

Here are some examples of knight moves:



Given a value for n , your program should write the number of valid n -key sequences on a single line to standard out.

Frequently Asked Questions

Here are the answers to some logistical questions that people sometimes have about completing this problem.

1. What language should I use?

We strongly suggest you use the Java language unless there's a language you feel stronger with.

2. What should I submit?

A zip file with the solution folder in it is probably the best thing. If you want to throw in a readme file to explain anything, a simple .txt file that can be opened with notepad or wordpad is best. Send the zip file as an attachment on a reply to this email message.

3. I use third party library X for everything that I do. Can I use it in this solution?

Sure. Please let us know that you have done that in your submission. If it's something that we can install (preferably for free) from the web, we will do that. If there is a redistributable dll that you want to include in your solution file, that's fine too, especially if it's important that we be on a common version of that library.

4. What will you do with my submission?

We will definitely compile and run it. We will also be spending a fair bit of time looking at the code itself and that may lead to some interesting conversations in later steps. We will attempt to run your solution at lengths 10, 16, and 32.

5. Is there a "right" solution?

The program should produce the correct number on the output, but failing to get the number right for interesting reasons is not a problem. Beyond getting the correct answer, we are interested in seeing how you program, so do what you think is best.

6. How much time do I have?

There are two answers for this. First, we would like to get the results back in a few days, so that we can all continue the process. However, if we have hit you at a really busy personal or work time, just let us know when you will be likely to get to it and we can adjust our plans. From a work-effort stand point, we do not impose any time limit.

7. Are there any tricky parts to understanding the problem?

We have tried to make the problem statement as simple as possible, but we have seen people get confused on a couple of points that might be worth clarifying, which are:

- It is OK for a key to be used multiple times in the same sequence, subject to the length and number of vowel constraints only. People familiar with the related-sounding "Knight's Tour" problem sometimes read in a requirement that the path never visit the same key twice. That is NOT a part of this problem.
- The only thing actually required on the output is the number of sequences. There are lots of ways to go at this problem and we want to constrain you as little as possible in getting to your solution.
- Some people are confused by the statement that paths 'do not wrap.' To clarify, what this means is that if a move goes off, say, the right edge of the board, it does not come back on the left. That is, the board is not on a cylinder or sphere or whatever. If a move would go off of the board, then it is not a legal move. It is not wrapped back onto the board.

8. Is there any way to tell if I have the right solution?

For path length of 10 with 2 vowels we expect the solution to contain all of the following digits at least once: 1, 3, 8, 9. If your solution does not contain at least one of each of these digits, then it is probably not exactly what we are looking for. You are correct in the observation that there is no “8” or “9” on the keypad. The code you write will produce a solution which is represented by a number. This number must include the digits 1, 3, 8 and 9 along with other digits and the 1, 3, 8 and 9 may be included more than once in the solution. Many candidates are confused by this so if you’d like to send the number of sequences you came up with (aka the solution) before submitting your code, please feel free to send and we will verify whether or not it is correct so if it’s not, you can rework your code.