Write a program to solve the “Vegetarians and Meat Eaters” problem.

Three vegetarians and three hungry meat-eaters need to cross a river. Unfortunately, the boat only holds two people. If the meat-eaters outnumber the vegetarians on either bank, the vegetarians will be eaten!

The **computer** must find a series of moves that gets all three vegetarians and all three meat-eaters across the river safely. I know you can solve the problem! Can the computer? You must give the computer some representation of the problem and the choices available. The computer must find the solution.

You are given broad leeway on how to represent this problem. Recursion and backtracking (depth-first search) is **one** way to solve this.

Good luck!

|  |
| --- |
| Example :    Start with three vegetarians and three meat-eaters on one side of the river.    Put one V and one M on the boat to cross.  This doesn’t change the count: there’s still three of each on the right side.    Have the boat cross.      Put one M in the boat.    Have the boat cross    There’s now two Vs and three Ms on one side of the river, so the vegetarians get eaten.  You can play the game at  https://www.numuki.com/game/cannibals-and-missionaries/ |

Hints:

* When there’s more meat-eaters than vegetarians on one side,   
  the vegetarian(s) still get eaten.  
  It doesn’t matter if one meat-eater stays in the boat or not.
* At least one person has to take the boat across the river

The base cases are:

4.1. 3 vegetarians and 3 meat eaters on left side of river = success  
4.2. 3 vegetarians and 3 meat eaters on right side of river = we’re back at the start and have looped around, stop following this path  
4.3. More meat eaters than vegetarians on left side of river = vegetarians get eaten  
4.4. More meat eaters than vegetarians on right side of river = vegetarians get eaten

Rubric:

* Student name and today’s date is a comment in the first line of the programs: -10 points if fails
* Screenshot and program code: -10 points if fails
* People start on *right* bank, end up on *left* bank: -10 points if fails
* Board state tracking/output: 5 points
* Moves change state of board: 5 points
* Computer moves search: 5 points   
  If the moves are hard-coded into the program, 0 points here.
* Program finds correct solution: 5 points  
  If the moves are hard-coded into the program, 0 points here.

Please paste a screenshot of a successful program run, and copy-and-paste the source code from your main program's .java file, here.