

Concepts of Programming Languages, Spring Term 2024
Project 2: Public Test Cases

1 createBoard

Note: Your create board function needs to output a valid board with all the correct peg positions and colors in the initial state. They do not have to be in the same order as the examples shown.

Test 1) > createBoard (-2,2)

Program error: The position is not valid.

Test 2) > createBoard (1,1)

```
[Peg (-3,-1) B,Peg (-3,0) B,Peg (-3,1) B,Peg (-2,-1) B,  
Peg (-2,0) B,Peg (-2,1) B,Peg (-1,-3) B,Peg (-1,-2) B,  
Peg (-1,-1) B,Peg (-1,0) B,Peg (-1,1) B,Peg (-1,2) B,  
Peg (-1,3) B,Peg (0,-3) B,Peg (0,-2) B,Peg (0,-1) B,  
Peg (0,0) B,Peg (0,1) B,Peg (0,2) B,Peg (0,3) B,  
Peg (1,-3) B,Peg (1,-2) B,Peg (1,-1) B,Peg (1,0) B,  
Peg (1,1) W,Peg (1,2) B,Peg (1,3) B,Peg (2,-1) B,  
Peg (2,0) B,Peg (2,1) B,Peg (3,-1) B,Peg (3,0) B,Peg (3,1) B]
```

Test 3) > createBoard (-2,1)

```
[Peg (-3,-1) B,Peg (-3,0) B,Peg (-3,1) B,Peg (-2,-1) B,  
Peg (-2,0) B,Peg (-2,1) W,Peg (-1,-3) B,Peg (-1,-2) B,  
Peg (-1,-1) B,Peg (-1,0) B,Peg (-1,1) B,Peg (-1,2) B,  
Peg (-1,3) B,Peg (0,-3) B,Peg (0,-2) B,Peg (0,-1) B,  
Peg (0,0) B,Peg (0,1) B,Peg (0,2) B,Peg (0,3) B,  
Peg (1,-3) B,Peg (1,-2) B,Peg (1,-1) B,Peg (1,0) B,  
Peg (1,1) B,Peg (1,2) B,Peg (1,3) B,Peg (2,-1) B,  
Peg (2,0) B,Peg (2,1) B,Peg (3,-1) B,Peg (3,0) B,Peg (3,1) B]
```

Test 4) > createBoard (1,-3)

```
[Peg (-3,-1) B,Peg (-3,0) B,Peg (-3,1) B,Peg (-2,-1) B,
Peg (-2,0) B,Peg (-2,1) B,Peg (-1,-3) B,Peg (-1,-2) B,
Peg (-1,-1) B,Peg (-1,0) B,Peg (-1,1) B,Peg (-1,2) B,
Peg (-1,3) B,Peg (0,-3) B,Peg (0,-2) B,Peg (0,-1) B,
Peg (0,0) B,Peg (0,1) B,Peg (0,2) B,Peg (0,3) B,
Peg (1,-3) W,Peg (1,-2) B,Peg (1,-1) B,Peg (1,0) B,
Peg (1,1) B,Peg (1,2) B,Peg (1,3) B,Peg (2,-1) B,
Peg (2,0) B,Peg (2,1) B,Peg (3,-1) B,Peg (3,0) B,Peg (3,1) B]
```

2 isValidMove

Test 5) > isValidMove (M (1,0))

```
[Peg (-3,-1) B,Peg (-3,0) B,Peg (-3,1) B,Peg (-2,-1) B,
Peg (-2,0) B,Peg (-2,1) B,Peg (-1,-3) B,Peg (-1,-2) B,
Peg (-1,-1) B,Peg (-1,0) B,Peg (-1,1) B,Peg (-1,2) B,
Peg (-1,3) B,Peg (0,-3) B,Peg (0,-2) B,Peg (0,-1) B,
Peg (0,0) B,Peg (0,1) B,Peg (0,2) B,Peg (0,3) B,
Peg (1,-3) B,Peg (1,-2) B,Peg (1,-1) B,Peg (1,0) B,
Peg (1,1) B,Peg (1,2) B,Peg (1,3) B,Peg (2,-1) B,
Peg (2,0) W,Peg (2,1) B,Peg (3,-1) B,Peg (3,0) B,Peg (3,1) B]
```

True

Test 6) > isValidMove (M (-2,0))

```
[Peg (-3,-1) B,Peg (-3,0) B,Peg (-3,1) B,Peg (-2,-1) B,
Peg (-2,0) B,Peg (-2,1) B,Peg (-1,-3) B,Peg (-1,-2) B,
Peg (-1,-1) B,Peg (-1,0) B,Peg (-1,1) B,Peg (-1,2) B,
Peg (-1,3) B,Peg (0,-3) B,Peg (0,-2) B,Peg (0,-1) B,
Peg (0,0) B,Peg (0,1) B,Peg (0,2) B,Peg (0,3) B,
Peg (1,-3) B,Peg (1,-2) B,Peg (1,-1) B,Peg (1,0) B,
Peg (1,1) B,Peg (1,2) B,Peg (1,3) B,Peg (2,-1) B,
Peg (2,0) W,Peg (2,1) B,Peg (3,-1) B,Peg (3,0) B,Peg (3,1) B]
```

False

explanation: The peg of that position does not have any white pegs around it so it cannot be flipped

```
Test 7) > isValidMove (M (-2,2))
[Peg (-3,-1) B,Peg (-3,0) B,Peg (-3,1) B,Peg (-2,-1) B,
Peg (-2,0) B,Peg (-2,1) B,Peg (-1,-3) B,Peg (-1,-2) B,
Peg (-1,-1) B,Peg (-1,0) B,Peg (-1,1) B,Peg (-1,2) B,
Peg (-1,3) B,Peg (0,-3) B,Peg (0,-2) B,Peg (0,-1) B,
Peg (0,0) B,Peg (0,1) B,Peg (0,2) B,Peg (0,3) B,
Peg (1,-3) B,Peg (1,-2) B,Peg (1,-1) B,Peg (1,0) B,
Peg (1,1) B,Peg (1,2) B,Peg (1,3) B,Peg (2,-1) B,
Peg (2,0) W,Peg (2,1) B,Peg (3,-1) B,Peg (3,0) B,Peg (3,1) B]
```

False

explanation: There is no peg on the board with position (-2,2)

```
Test 8) > isValidMove (M (0,0))
[Peg (-3,-1) B,Peg (-3,0) B,Peg (-3,1) B,Peg (-2,-1) B,
Peg (-2,0) B,Peg (-2,1) B,Peg (-1,-3) B,Peg (-1,-2) B,
Peg (-1,-1) W,Peg (-1,0) B,Peg (-1,1) W,Peg (-1,2) B,
Peg (-1,3) B,Peg (0,-3) B,Peg (0,-2) B,Peg (0,-1) B,
Peg (0,0) B,Peg (0,1) B,Peg (0,2) B,Peg (0,3) B,
Peg (1,-3) B,Peg (1,-2) B,Peg (1,-1) W,Peg (1,0) B,
Peg (1,1) W,Peg (1,2) B,Peg (1,3) B,Peg (2,-1) B,
Peg (2,0) B,Peg (2,1) B,Peg (3,-1) B,Peg (3,0) B,Peg (3,1) B]
```

False

explanation: The only white pegs around the position are on the diagonal

```
Test 9) > isValidMove (M (2,-1))
[Peg (-3,-1) B,Peg (-3,0) B,Peg (-3,1) B,Peg (-2,-1) B,
Peg (-2,0) B,Peg (-2,1) B,Peg (-1,-3) B,Peg (-1,-2) B,
Peg (-1,-1) B,Peg (-1,0) B,Peg (-1,1) B,Peg (-1,2) B,
Peg (-1,3) B,Peg (0,-3) B,Peg (0,-2) B,Peg (0,-1) B,
Peg (0,0) B,Peg (0,1) B,Peg (0,2) B,Peg (0,3) B,
Peg (1,-3) B,Peg (1,-2) B,Peg (1,-1) W,Peg (1,0) B,
Peg (1,1) B,Peg (1,2) B,Peg (1,3) B,Peg (2,-1) B,
Peg (2,0) B,Peg (2,1) B,Peg (3,-1) W,Peg (3,0) B,Peg (3,1) B]
```

True

3 isGoal

Test 10) > isGoal

```
[Peg (-3,-1) W,Peg (-3,0) W,Peg (-3,1) W,Peg (-2,-1) W,  
Peg (-2,0) W,Peg (-2,1) W,Peg (-1,-3) W,Peg (-1,-2) W,  
Peg (-1,-1) W,Peg (-1,0) W,Peg (-1,1) W,Peg (-1,2) W,  
Peg (-1,3) W,Peg (0,-3) W,Peg (0,-2) W,Peg (0,-1) W,  
Peg (0,0) W,Peg (0,1) W,Peg (0,2) W,Peg (0,3) W,  
Peg (1,-3) W,Peg (1,-2) W,Peg (1,-1) W,Peg (1,0) W,  
Peg (1,1) W,Peg (1,2) W,Peg (1,3) W,Peg (2,-1) W,  
Peg (2,0) W,Peg (2,1) W,Peg (3,-1) W,Peg (3,0) W,Peg (3,1) W]
```

True

Test 11) > isGoal

```
[Peg (-3,-1) W,Peg (-3,0) W,Peg (-3,1) W,Peg (-2,-1) W,  
Peg (-2,0) W,Peg (-2,1) W,Peg (-1,-3) W,Peg (-1,-2) W,  
Peg (-1,-1) W,Peg (-1,0) W,Peg (-1,1) W,Peg (-1,2) W,  
Peg (-1,3) W,Peg (0,-3) W,Peg (0,-2) W,Peg (0,-1) W,  
Peg (0,0) W,Peg (0,1) B,Peg (0,2) W,Peg (0,3) W,  
Peg (1,-3) W,Peg (1,-2) W,Peg (1,-1) W,Peg (1,0) W,  
Peg (1,1) W,Peg (1,2) W,Peg (1,3) W,Peg (2,-1) W,  
Peg (2,0) W,Peg (2,1) W,Peg (3,-1) W,Peg (3,0) W,Peg (3,1) W]
```

False

Test 12) > isGoal

```
[Peg (2,-1) W,Peg (1,-3) W,Peg (0,-1) W,Peg (2,0) W,  
Peg (1,1) W,Peg (1,-1) W,Peg (0,2) W,Peg (-1,-3) W,  
Peg (-3,1) W,Peg (0,0) W,Peg (-3,0) W,Peg (-1,2) W,  
Peg (-1,-2) W,Peg (-1,1) W,Peg (1,-2) W,Peg (1,0) W,  
Peg (0,-2) W,Peg (-2,-1) W,Peg (-1,-1) W,Peg (3,-1) W,  
Peg (-2,1) W,Peg (-3,-1) W,Peg (3,1) W,Peg (0,-3) W,  
Peg (1,3) W,Peg (2,1) W,Peg (0,1) W,Peg (1,2) W,Peg (0,3) W,  
Peg (-2,0) W,Peg (-1,3) W,Peg (-1,0) W,Peg (3,0) W]
```

True

4 showPossibleNextStates

Test 13) > showPossibleNextStates

```
[Peg (-3,-1) W,Peg (-3,0) W,Peg (-3,1) W,Peg (-2,-1) W,  
Peg (-2,0) W,Peg (-2,1) W,Peg (-1,-3) W,Peg (-1,-2) W,  
Peg (-1,-1) W,Peg (-1,0) W,Peg (-1,1) W,Peg (-1,2) W,  
Peg (-1,3) W,Peg (0,-3) W,Peg (0,-2) W,Peg (0,-1) W,  
Peg (0,0) W,Peg (0,1) B,Peg (0,2) W,Peg (0,3) W,  
Peg (1,-3) W,Peg (1,-2) W,Peg (1,-1) W,Peg (1,0) W,  
Peg (1,1) W,Peg (1,2) W,Peg (1,3) W,Peg (2,-1) W,  
Peg (2,0) W,Peg (2,1) W,Peg (3,-1) W,Peg (3,0) W,Peg (3,1) W]
```

```
[S (M (0,1)) [Peg (-3,-1) W,Peg (-3,0) W,
Peg (-3,1) W,Peg (-2,-1) W,Peg (-2,0) W,Peg (-2,1) W,
Peg (-1,-3) W,Peg (-1,-2) W,Peg (-1,-1) W,Peg (-1,0) W,
Peg (-1,1) W,Peg (-1,2) W,Peg (-1,3) W,Peg (0,-3) W,
Peg (0,-2) W,Peg (0,-1) W,Peg (0,0) W,Peg (0,1) W,
Peg (0,2) W,Peg (0,3) W,Peg (1,-3) W,Peg (1,-2) W,
Peg (1,-1) W,Peg (1,0) W,Peg (1,1) W,Peg (1,2) W,
Peg (1,3) W,Peg (2,-1) W,Peg (2,0) W,Peg (2,1) W,
Peg (3,-1) W,Peg (3,0) W,Peg (3,1) W]]
```

explanation: The state shown in this example is the only possible one in this case because the only valid possible move is to change the piece at (0,1) to be white.

Test 14) > showPossibleNextStates

```
[Peg (-3,-1) W,Peg (-3,0) W,Peg (-3,1) W,Peg (-2,-1) W,
Peg (-2,0) W,Peg (-2,1) W,Peg (-1,-3) W,Peg (-1,-2) W,
Peg (-1,-1) W,Peg (-1,0) W,Peg (-1,1) W,Peg (-1,2) W,
Peg (-1,3) W,Peg (0,-3) W,Peg (0,-2) W,Peg (0,-1) W,
Peg (0,0) W,Peg (0,1) W,Peg (0,2) W,Peg (0,3) W,
Peg (1,-3) W,Peg (1,-2) W,Peg (1,-1) W,Peg (1,0) W,
Peg (1,1) W,Peg (1,2) W,Peg (1,3) W,Peg (2,-1) W,
Peg (2,0) W,Peg (2,1) W,Peg (3,-1) W,Peg (3,0) W,Peg (3,1) W]
```

Program error: No Possible States Exist.

explanation: There are no possible states since there are no valid moves to be made.

Test 15) > showPossibleNextStates

```
[Peg (-3,-1) W,Peg (-3,0) W,Peg (-3,1) W,Peg (-2,-1) W,
Peg (-2,0) W,Peg (-2,1) W,Peg (-1,-3) W,Peg (-1,-2) W,
Peg (-1,-1) W,Peg (-1,0) B,Peg (-1,1) W,Peg (-1,2) W,
Peg (-1,3) W,Peg (0,-3) W,Peg (0,-2) W,Peg (0,-1) B,
Peg (0,0) B,Peg (0,1) W,Peg (0,2) W,Peg (0,3) W,
Peg (1,-3) W,Peg (1,-2) W,Peg (1,-1) W,Peg (1,0) B,
Peg (1,1) W,Peg (1,2) W,Peg (1,3) W,Peg (2,-1) W,Peg (2,0) W,
Peg (2,1) W,Peg (3,-1) W,Peg (3,0) W,Peg (3,1) W]
```

```
[
S (M (-1,0)) [Peg (-3,-1) W,Peg (-3,0) W,Peg (-3,1) W,Peg (-2,-1) W,
Peg (-2,0) W,Peg (-2,1) W,Peg (-1,-3) W,Peg (-1,-2) W,Peg (-1,-1) W,
Peg (-1,0) W,Peg (-1,1) W,Peg (-1,2) W,Peg (-1,3) W,Peg (0,-3) W,
Peg (0,-2) W,Peg (0,-1) B,Peg (0,0) B,Peg (0,1) W,Peg (0,2) W,
Peg (0,3) W,Peg (1,-3) W,Peg (1,-2) W,Peg (1,-1) W,Peg (1,0) B,
Peg (1,1) W,Peg (1,2) W,Peg (1,3) W,Peg (2,-1) W,
Peg (2,0) W,Peg (2,1) W,Peg (3,-1) W,Peg (3,0) W,Peg (3,1) W]
,
```

```

S (M (0,0)) [Peg (-3,-1) W,Peg (-3,0) W,Peg (-3,1) W,Peg (-2,-1) W,
Peg (-2,0) W,Peg (-2,1) W,Peg (-1,-3) W,Peg (-1,-2) W,Peg (-1,-1) W,
Peg (-1,0) B,Peg (-1,1) W,Peg (-1,2) W,Peg (-1,3) W,Peg (0,-3) W,
Peg (0,-2) W,Peg (0,-1) B,Peg (0,0) W,Peg (0,1) W,Peg (0,2) W,
Peg (0,3) W,Peg (1,-3) W,Peg (1,-2) W,Peg (1,-1) W,Peg (1,0) B,
Peg (1,1) W,Peg (1,2) W,Peg (1,3) W,Peg (2,-1) W,Peg (2,0) W,
Peg (2,1) W,Peg (3,-1) W,Peg (3,0) W,Peg (3,1) W]
,
S (M (0,-1)) [Peg (-3,-1) W,Peg (-3,0) W,Peg (-3,1) W,Peg (-2,-1) W,
Peg (-2,0) W,Peg (-2,1) W,Peg (-1,-3) W,Peg (-1,-2) W,Peg (-1,-1) W,
Peg (-1,0) B,Peg (-1,1) W,Peg (-1,2) W,Peg (-1,3) W,Peg (0,-3) W,
Peg (0,-2) W,Peg (0,-1) W,Peg (0,0) B,Peg (0,1) W,Peg (0,2) W,
Peg (0,3) W,Peg (1,-3) W,Peg (1,-2) W,Peg (1,-1) W,Peg (1,0) B,
Peg (1,1) W,Peg (1,2) W,Peg (1,3) W,Peg (2,-1) W,Peg (2,0) W,
Peg (2,1) W,Peg (3,-1) W,Peg (3,0) W,Peg (3,1) W]
,
S (M (1,0)) [Peg (-3,-1) W,Peg (-3,0) W,Peg (-3,1) W,Peg (-2,-1) W,
Peg (-2,0) W,Peg (-2,1) W,Peg (-1,-3) W,Peg (-1,-2) W,Peg (-1,-1) W,
Peg (-1,0) B,Peg (-1,1) W,Peg (-1,2) W,Peg (-1,3) W,Peg (0,-3) W,
Peg (0,-2) W,Peg (0,-1) B,Peg (0,0) B,Peg (0,1) W,Peg (0,2) W,
Peg (0,3) W,Peg (1,-3) W,Peg (1,-2) W,Peg (1,-1) W,Peg (1,0) W,
Peg (1,1) W,Peg (1,2) W,Peg (1,3) W,Peg (2,-1) W,Peg (2,0) W,
Peg (2,1) W,Peg (3,-1) W,Peg (3,0) W,Peg (3,1) W]
]

```

explanation: There are 4 possible moves which so there are 4 possible states

Test 16) > showPossibleNextStates

```

[Peg (-3,-1) W,Peg (-3,0) W,Peg (-3,1) W,Peg (-2,-1) W,
Peg (-2,0) W,Peg (-2,1) W,Peg (-1,-3) W,Peg (-1,-2) W,
Peg (-1,-1) W,Peg (-1,0) W,Peg (-1,1) W,Peg (-1,2) W,
Peg (-1,3) W,Peg (0,-3) W,Peg (0,-2) W,Peg (0,-1) W,
Peg (0,0) W,Peg (0,1) W,Peg (0,2) W,Peg (0,3) W,
Peg (1,-3) W,Peg (1,-2) W,Peg (1,-1) W,Peg (1,0) W,
Peg (1,1) W,Peg (1,2) W,Peg (1,3) W,Peg (2,-1) B,Peg (2,0) B,
Peg (2,1) B,Peg (3,-1) B,Peg (3,0) B,Peg (3,1) B]

[
S (M (2,-1)) [Peg (-3,-1) W,Peg (-3,0) W,Peg (-3,1) W,Peg (-2,-1) W,
Peg (-2,0) W,Peg (-2,1) W,Peg (-1,-3) W,Peg (-1,-2) W,Peg (-1,-1) W,
Peg (-1,0) W,Peg (-1,1) W,Peg (-1,2) W,Peg (-1,3) W,Peg (0,-3) W,
Peg (0,-2) W,Peg (0,-1) W,Peg (0,0) W,Peg (0,1) W,Peg (0,2) W,
Peg (0,3) W,Peg (1,-3) W,Peg (1,-2) W,Peg (1,-1) W,Peg (1,0) W,
Peg (1,1) W,Peg (1,2) W,Peg (1,3) W,Peg (2,-1) W,Peg (2,0) B,
Peg (2,1) B,Peg (3,-1) B,Peg (3,0) B,Peg (3,1) B]
,

```

```

S (M (2,0)) [Peg (-3,-1) W,Peg (-3,0) W,Peg (-3,1) W,Peg (-2,-1) W,
Peg (-2,0) W,Peg (-2,1) W,Peg (-1,-3) W,Peg (-1,-2) W,Peg (-1,-1) W,
Peg (-1,0) W,Peg (-1,1) W,Peg (-1,2) W,Peg (-1,3) W,Peg (0,-3) W,
Peg (0,-2) W,Peg (0,-1) W,Peg (0,0) W,Peg (0,1) W,Peg (0,2) W,
Peg (0,3) W,Peg (1,-3) W,Peg (1,-2) W,Peg (1,-1) W,Peg (1,0) W,
Peg (1,1) W,Peg (1,2) W,Peg (1,3) W,Peg (2,-1) B,Peg (2,0) W,
Peg (2,1) B,Peg (3,-1) B,Peg (3,0) B,Peg (3,1) B]
,
S (M (2,1)) [Peg (-3,-1) W,Peg (-3,0) W,Peg (-3,1) W,Peg (-2,-1) W,
Peg (-2,0) W,Peg (-2,1) W,Peg (-1,-3) W,Peg (-1,-2) W,Peg (-1,-1) W,
Peg (-1,0) W,Peg (-1,1) W,Peg (-1,2) W,Peg (-1,3) W,Peg (0,-3) W,
Peg (0,-2) W,Peg (0,-1) W,Peg (0,0) W,Peg (0,1) W,Peg (0,2) W,
Peg (0,3) W,Peg (1,-3) W,Peg (1,-2) W,Peg (1,-1) W,Peg (1,0) W,
Peg (1,1) W,Peg (1,2) W,Peg (1,3) W,Peg (2,-1) B,Peg (2,0) B,
Peg (2,1) W,Peg (3,-1) B,Peg (3,0) B,Peg (3,1) B]
]

```

explanation: There are 3 possible moves which so there are 3 possible states