Assignment 8b - Recursion

1. Take as input N, a number. N is the size of a snakes and ladder board (without any snakes and ladders). Take as input M, a number. M is the number of faces of the dice.
    a. Write a recursive function which returns the count of different ways the board can be travelled using the dice. Print the value returned.
    b. Write a recursive function which returns an ArrayList of dice values for all valid paths across the board. Print the value returned.
    e.g. for a board of size 10 and dice size 6, a few valid paths will be "64", "46", "2332", "541" etc.
    c. Write a recursive function which prints dice-values for all valid paths across the board (void is the return type for function).
2. Take as input N1 and N2, both numbers. N1 and N2 is the number of rows and columns on a rectangular board. Our player starts in top-left corner of the board and must reach bottom-right corner. In one move the player can move 1 step horizontally (right) or 1 step vertically (down).
    a. Write a recursive function which returns the count of different ways the player can travel across the board. Print the value returned.
    b. Write a recursive function which returns an ArrayList of moves for all valid paths across the board. Print the value returned.
    e.g. for a board of size 3,3; a few valid paths will be "HHVV", "VVHH", "VHHV" etc.
    c. Write a recursive function which prints moves for all valid paths across the board (void is the return type for function).
3. Take as input N1 and N2, both numbers. N1 and N2 is the number of rows and columns on a rectangular board. Our player starts in top-left corner of the board and must reach bottom-right corner. In one move the player can move 1 step horizontally (right) or 1 step vertically (down) or 1 step diagonally (south-east).
    a. Write a recursive function which returns the count of different ways the player can travel across the board. Print the value returned.
    b. Write a recursive function which returns an ArrayList of moves for all valid paths across the board. Print the value returned.
    e.g. for a board of size 3,3; a few valid paths will be "HHVV", "VVHH", "VHHV", "DD", "VDH", "HVD" etc.
    c. Write a recursive function which prints moves for all valid paths across the board (void is the return type for function).
4. Take as input N1 and N2, both numbers. N1 and N2 is the number of rows and columns on a rectangular board. Our player starts in top-left corner of the board and must reach bottom-right corner. In one move the player can move 1 step horizontally (right) or 1 step vertically (down) or 1 step diagonally (south-east). But

Assignment 8b - Recursion

the diagonal step is allowed only when the player is currently on one of the diagonals (there are two diagonals)

   a. Write a recursive function which returns the count of different ways the player can travel across the board. Print the value returned.
   b. Write a recursive function which returns an ArrayList of moves for all valid paths across the board. Print the value returned.
      E.g. for a board of size 3,3; a few valid paths will be "HHVV", "VVHH", "VHHV", "DD", "VHD", "HVD" etc.
   c. Write a recursive function which prints moves for all valid paths across the board (void is the return type for function).

5. Take as input N, the size of a chess board. We are asked to place N number of queens in it, so that no queen can kill other.
   a. Write a recursive function which returns the count of different distinct ways the queens can be placed across the board. Print the value returned.
   b. Write a recursive function which returns an ArrayList of all valid configurations. Print the value returned.
      E.g. for a board of size 4; there are two possible configurations "{0-1}, {1-3}, {2-0}, {3-2}" and "{0-2}, {1-0}, {2-3}, {3-1}"
   c. Write a recursive function which prints all valid configurations (void is the return type for function).
6. Take as input N, the size of a chess board. We are asked to place N number of Knights in it, so that no knight can kill other.
   a. Write a recursive function which returns the count of different distinct ways the knights can be placed across the board. Print the value returned.
   b. Write a recursive function which returns an ArrayList of all valid configurations. Print the value returned.
   c. Write a recursive function which prints all valid configurations (void is the return type for function).
7. Take as input N, a number. N is the size of a snakes and ladders board. There are no snakes but we've ladders from 1st prime number to last prime number in range, 2nd prime number to 2nd from last prime number and so-on.
   a. Write a recursive function which returns the count of different distinct ways this board can be crossed with a normal dice. Print the value returned.
   b. Write a recursive function which returns an ArrayList of all valid paths (in term of dice values). Print the value returned.
   c. Write a recursive function which prints all valid paths (void is the return type for function).
8. Take as input N, a number. N is the size of a snakes and ladders board. We've chosen the following strategy to place snakes and ladders on the board –
   a. From lowest to highest prime number there is a ladder

Assignment 8b - Recursion

b. From 2nd highest to 2nd lowest prime number there is a snake
E.g. for a board of size 20, the prime numbers are 2, 3, 5, 7, 11, 13, 17, 19
Ladders are from 2 to 19 and 5 to 13
Snakes are from 17 to 3 and 11 to 7

Now take as input M, another number. Take input M inputs now which represent the values on dice for M throws.
Write a recursive function which checks if the board (with its snakes and ladders) can be crossed using the M dice values and returns a Boolean value. Print the value returned.

9. Take as input N, a number. N represents the size of a chess board. We've a piece standing in top-left corner and it must reach the bottom-right corner. The piece moves as follows –
   a. In any cell, the piece moves like a knight. But out of the possible 8 moves for a knight only the positive ones are valid i.e. both row and column must increase in a move.
   b. On the walls (4 possible walls), the piece can move like a rook as well (in addition of knight moves). But, only the positive moves are allowed i.e. as a rook, piece can move any number of steps horizontally or vertically but in a manner, such that row or column must increase.
   c. On the diagonals (2 possible diagonals), the piece can move like a bishop as well (in addition to the knight and possibly rook moves). But, only the positive moves are allowed i.e. as a bishop, piece can move in a way such that row and column must increase.

   You are supposed to write the following functions
   a. Write a recursive function which returns the count of different distinct ways this board can be crossed. Print the value returned.
   b. Write a recursive function which returns an ArrayList of all valid. Print the value returned. E.g. Following are two valid paths for a board of size 4
   {0-0}, {2-1}, {3-3}
   {0-0}, {1-2}, {3-3}
   c. Write a recursive function which prints all valid paths (void is the return type for function).
10. Take as input N, a number. N represents the size of a chess board. The cells in board are numbered. The top-left cell is numbered 1 and numbering increases from left to right and top to bottom. E.g.

1	2	3	4
5	6	7	8
9	10	11	12

Assignment 8b - Recursion

13	14	15	16

Prime numbers act as mines and ports alternately i.e. first prime number is a mine while second is a port and so on. Piece can go over a mine but cannot stop on it. Piece can directly move from a port to the destination (but may not chose to).

We've a piece standing in top-left corner and it must reach the bottom-right corner. The piece moves as follows –

a. In any cell, the piece moves like a knight. But out of the possible 8 moves for a knight only the positive ones are valid i.e. both row and column must increase in a move.
b. On the walls (4 possible walls), the piece can move like a rook as well (in addition of knight moves). But, only the positive moves are allowed i.e. as a rook, piece can move any number of steps horizontally or vertically but in a manner, such that row or column must increase.
c. On the diagonals (2 possible diagonals), the piece can move like a bishop as well (in addition to the knight and possibly rook moves). But, only the positive moves are allowed i.e. as a bishop, piece can move in a way such that row and column must increase.

You are supposed to write the following functions

a. Write a recursive function which returns the count of different distinct ways this board can be crossed. Print the value returned.
b. Write a recursive function which returns an ArrayList of all valid. Print the value returned. E.g. Following are two valid paths for a board of size 4
   {0-0}, {2-1}, {3-3}
   {0-0}, {0-2}, {3-3}
c. Write a recursive function which prints all valid paths (void is the return type for function).