```
class Board:
    def __init__(self, rows, cols):
        .....
        Board constructor
        .....
        . . .
    def get_closest_food(self,bear):
        .....
        Return Location of food closest to bear.
        .....
        . . .
    def get_all_food_locs(self):
        .....
        Returns a list of the Locations of food on the board.
        .....
        . . .
    def is_legal_move(self, proposed_move):
        .....
        Returns True if the square proposed_move is blank, False otherwise.
        .....
        . . .
class Location:
    def __init__(self, x, y):
        .....
        Location constructor
        .....
        . . .
    def distance(self, other):
        .....
        Gives the manhatten distance between self and other
        .....
        . . .
    def is_adjacent(self, other):
        .....
        returns True if self and other are next to each other, False otherwise.
        .....
        . . .
    def ___eq__(self, other):
        .....
        This special function is used when comparing two distances to one
        another. It should return True if the x and y values of self and
        other are the same, False if they are not
        .....
        . . .
class Bear:
    def init (self, loc = Location(0, 0), energy = 0):
        .....
        Bear constructor
        .....
        . . .
```

def eat_food(self, board, food_location): The Bear moves to food_location on board and eats the food there. Assumes that the bear is next to food location and that food location does in fact contain food def move_towards_food(self, board, food_loc): Moves the bear in the direction of food_loc. Returns True if the bear does move, False otherwise def move_one_square(self, board, toward): moves the bear one square closer to toward. If for whatever reason the bear is unable to move, returns False. Otherwise returns True def setup(rows, cols): Makes a board of rows by cols, places a bear randomly in a legal position on the board and then starts the simulation def play(board, bear): Runs the simulation involving Board board and Bear bear. Every step of the simulation involves the bear moving one square. The simulation ends when the bear runs out of food to eat. def print state (board, bear): Prints out board, including the location of all the food and the location of the bear.