Hi everyone. This is for anyone who might be struggling to understand my implementation. Let's break down this Hospital Karel program line by line so that you understand exactly what's happening:

The Program Setup

```
from karel.stanfordkarel import *
"""
Program: Hospital Karel
Karel traverses 1st Street from west to east, building hospitals
wherever it encounters a beeper.
"""
```

- The text in triple quotes (`"""`) is called a "docstring" - it's just a description of what our program does. The computer ignores this part.
- `from karel.stanfordkarel import *` tells the computer to load all the special Karel commands we'll need. Think of this as getting Karel's instruction manual ready.

The Main Function

```
def main():
    # Check the first position before starting to move
    if beepers_present():
        build_hospital()

    # Move along the first row, checking each position
    while front_is_clear():
        move()
        if beepers_present():
            build_hospital()
```

- `def main():` creates our main function - this is where Karel's journey begins.
- `if beepers_present():` asks "Hey Karel, is there a beeper where you're standing right now?"
    - If yes, then we run `build_hospital()` (we'll explain this function later)
- `while front_is_clear():` creates a loop that continues as long as there's no wall in front of Karel
    - Think of this as saying "Keep doing the following steps until you hit a wall"
- Inside that loop:

    - `move()` tells Karel to move forward one step
    - `if beepers_present():` checks again if Karel is standing on a beeper
    - If yes, then `build_hospital()` runs and Karel builds a hospital

This main function handles Karel's entire journey across the first row, stopping to build hospitals whenever it finds beepers.

The Build Hospital Function

```
def build_hospital():
    """
    Karel picks up supplies and builds a hospital.
    Pre-condition: Karel is on a beeper, representing a
        pile of supplies. Karel is facing east.
    Post-condition: Karel is standing at the bottom
        of the last column of the hospital, facing east.
    """
    # pick up supplies
    pick_beeper()

    # Build first column upward
    turn_left()  # Face north
    put_three_beepers()

    # Move to top of second column
    turn_right()  # Face east
    move()

    # Build second column downward
    turn_right()  # Face south
    put_three_beepers()

    # Already at the bottom of second column, facing south
    turn_left()  # Face east for the next hospital
```

- This function handles the entire hospital-building process
- `pick_beeper()` makes Karel pick up the supply beeper
- `turn_left()` rotates Karel so he faces north (upward)
- `put_three_beepers()` is a helper function that makes Karel build one column of the hospital (we'll explain this function next)
- After building the first column, Karel is at the top facing north
- `turn_right()` makes Karel face east (to the right)
- `move()` moves Karel one step east to the top of where the second column should be
- Another `turn_right()` makes Karel face south (downward)
- Another call to `put_three_beepers()` makes Karel build the second column, going downward
- After building the second column, Karel is at the bottom facing south
- One more `turn_left()` makes Karel face east again, ready to continue his journey

The Put Three Beepers Function

```
def put_three_beepers():
    """
    Karel places three beepers in a row.
    Pre-condition: Karel is on the corner where we want
        to place the first beeper.
    Post-condition: Karel is on the corner where it
        placed the third beeper in a row.
    """
    put_beeper()
    move()
    put_beeper()
    move()
    put_beeper()
```

- This function makes Karel put down three beepers in a straight line
- `put_beeper()` places a beeper at Karel's current position
- `move()` makes Karel move one step forward
- `put_beeper()` places another beeper
- `move()` makes Karel move one more step
- `put_beeper()` places the third and final beeper
- Notice there's no final `move()` - Karel stays on the last position

The Turn Right Function

```
def turn_right():
    """Turn Karel 90 degrees to the right"""
    turn_left()
    turn_left()
    turn_left()
```

- Karel only knows how to turn left, but we need him to turn right sometimes
- This function uses three left turns to make a right turn
- Think of it like this: if you turn left three times, you end up facing right relative to where you started

Program Execution

```
if __name__ == '__main__':
    main()
```

- This special code tells the computer "When you run this program, start with the main function"
- It's like the "ON" switch for our Karel program

Walking Through an Example

Imagine Karel is on the bottom-left corner of a grid with beepers at positions 3 and 7:

1. Karel checks if there's a beeper at his starting position (there isn't)
2. Karel enters the while loop and starts moving east
3. After 2 moves, Karel reaches position 3 and finds a beeper
4. Karel builds a hospital:
    - Picks up the beeper
    - Turns north and builds the first column upward
    - Moves to the top of the second column
    - Turns south and builds the second column downward
    - Turns east to continue
5. Karel continues moving east
6. At position 7, Karel finds another beeper and builds another hospital
7. Karel continues until hitting a wall, at which point the program ends

This way, Karel builds hospitals at each location where supplies (beepers) are available!