Lab 4: Text Adventure

- Make a text adventure game
- Player is in a room inside a dungeon
- The player can move to new rooms using compass directions
- When a new room is entered, the room description is printed

```
This is an open field west of a white house, with
a boarded front door.
$ north
North of House: You are facing the north side of a
white house.
$ east
Behind House: You are behind the white house.
$ south
```

The Dungeon

- The dungeon is a set of interconnected rooms
- Each room has a description and a set of neighboring rooms
  - One neighbor in each compass direction
- The dungeon must be described in a **dungeon file** which is read by your program
- Your program must accept any properly formatted dungeon file
- **`loaddungeon`** command is used to load a dungeon file

```
$ loaddungeon dfile.txt
This is an open field west of a white house, with
a boarded front door.
```

Dungeon File Format

- Each line of the file contains information about one room
- There are no blank lines, so the number of lines is the number of rooms
- Each room is described by 6 item of information:

1. ***Room number***: This is *any* positive integer which uniquely identifies the room.
2. ***Description***: This is a string which is printed when the player enters the room.
3. ***North room***: This is the room number of the room immediately to the north of this room. If there is no room to the north of this room then this value is -1.
4. ***South room***: This is the room number of the room immediately to the south of this room. If there is no room to the south of this room then this value is -1.
5. ***East room***: This is the room number of the room immediately to the east of this room. If there is no room to the east of this room then this value is -1.
6. ***West room***: This is the room number of the room immediately to the west of this room. If there is no room to the west of this room then this value is -1

Example Dungeon File

```
1 “This is an open field west of a white house, with a boarded front door.” 2 -1 -1 -1
2 “North of House: You are facing the north side of a white house.” -1 1 3 -1
3 “Behind House: You are behind the white house.” -1 4 -1 2
4 “South of House: You are facing the south side of a white house.” 3 -1 -1 -1
```

- Each pair of items on a line is separated by one or more spaces
- There is a newline character (‘\n’) at the end of each line
- The room numbers do not need to start at 1
- The room numbers do not need to be sequential

Commands

- `loaddungeon`: This command loads the information contained in a dungeon file. The command takes one argument, the name of the dungeon file. This command should be the first command issued by the player after starting the game.
- `north`: This command moves the player into the room to the north of the current room.
- `south`: This command moves the player into the room to the south of the current room.
- `east`: This command moves the player into the room to the east of the current room.
- `west`: This command moves the player into the room to the west of the current room.
- `quit`: This command should end the program

Exception Behaviors

- Your shell should be case-sensitive
- Ignore any invalid commands, and just print a new prompt
- If `loaddungeon` is called with incorrect arguments, ignore the command and print a new prompt
- If a user tries to move in a direction which is not connected to a room, print "You can't go there" and do not change the room.

Programming Requirements

- Be sure to define a function called `StartDungeon()` which starts the execution of your code. At the bottom of your file you should call `StartDungeon()` to begin running your code.
- Include comments for each function
  - Docsting
  - Input/Output definition
  - Label types of function arguments and return value
- Create a namedtuple to describe each room.
- Store all of the room namedtuples in a dictionary and use the room numbers as the keys.