FanDuel Coding Challenge

Please answer the below question to the best of your ability. If you have any questions do not hesitate to reach out to us for clarification. You may use any language/framework you feel comfortable with to implement your solution.

Please provide us with all files and instructions needed to build and/or run your code. Instructions should be placed inside a README.md file at the project root level.

1) Depth Charts

Most team sports have a depth chart (a ranking of each player) for each position they have. For Example in NFL: Ben Roethlisberger is listed as the starting QB and first on the QB depth chart. Landry Jones, his backup is listed as the 2nd person on that depth chart. We want to implement functionality that will manage these depth charts.

Data Model

Assume player objects look like this. Note that players can be on the depth chart for positions that are not their own.

```
{
"player_id": 1,
"name": "Bob",
"position": "WR"
}
```

Use Cases to Implement

- **`addPlayerToDepthChart(player, position, position_depth)`**
  Adds a player to a depth chart for a given position (at a specific spot). If no `position_depth` is provided, then add them to the end of the depth chart for that position. If you are entering two players into the same slot, the last player entered gets priority and bumps the existing player down a depth spot.

- **`removePlayerFromDepthChart(player, position)`**
  Removes a player from the depth chart for a position

- **`getFullDepthChart()`**
  Prints out all depth chart positions

- **`getPlayersUnderPlayerInDepthChart(player, position)`**
  For a given player find all players below them on the depth chart.

Example

```
var bob = { "player_id": 1, "name": "Bob" }
var alice = { "player_id": 2, "name": "Alice" }
var charlie = { "player_id": 3, "name": "Charlie"}

addPlayerToDepthChart(bob, "WR", 0);
addPlayerToDepthChart(alice, "WR", 0);
addPlayerToDepthChart(charlie, "WR", 2);
addPlayerToDepthChart(bob, "KR");
getFullDepthChart();

/*
Output:
WR: [2, 1, 3],
KR: [1]
*/

getPlayersUnderPlayerInDepthChart(alice, "WR");

/*
Output:
      [1,3]
*/
```

Please implement the 4 use cases above for:

1. NFL supporting positions (QB, WR, RB, TE, K, P, KR, PR)
2. MLB supporting positions (SP, RP, C, 1B, 2B, 3B, SS, LF, RF, CF, DH).

Follow Up

Keep in mind we potentially might add more sports in the future and we want to make it as easy as possible to add new ones.

2) March Madness Bracket

Please code the methods that would go into creating a filled out March Madness Bracket.

Specifically:

1. Build out the classes and fields that you would use to construct this.
2. Add functionality to fill out the bracket (via command line prompt, seed file, or method of your choice). Functionality should include the ability to pick/seed the teams in the tournament as well as picking the winner for any game in any order.
3. Implement the following use cases:
    1. **`seedTeam(team, seed, region)`**
       Adds a team to a specific seed in a specific region

    2. **`advanceTeam(team)`**
       Chooses a team to advance to their next possible game

    3. **`isBracketComplete()`**
       Returns if a bracket is fully filled out or not

    4. **`findChampion()`**
       Given a completed bracket, returns the winner/champion

    5. **`championsPathToVictory()`**
       Given a completed bracket, returns the teams the champion beat in order starting from the first round

NOTE: You do not need to use these exact method signatures, but you need to have methods that satisfy that described use case.

Please feel free to reach out with any/all clarifying questions.

If you're curious about March Madness, please refer to the following links:

- https://en.wikipedia.org/wiki/March_Madness_pools
- https://www.ncaa.com/news/basketball-men/bracketiq/2018-03-13/what-march-madness-ncaa-tournament-explained
- You do not need to model the play-in games (only the top 64 teams), but you can as an additional challenge

Sample Output

A sample output for the methods

```
isBracketComplete()
// "Bracket is complete"
findChampion()
// "Your champion is Penn"
championsPathToVictory()
// "Your champions path to victory was Kansas -> Seton Hall ->
```

Here is a sample of how a command line interface would work:

```
What Team is Seed 1 in the SOUTH region?
Virginia
What Team is Seed 2 in the SOUTH region?
Cincinnati
What Team is Seed 3 in the SOUTH region?
Tennessee
What Team is Seed 4 in the SOUTH region?
Arizona
What Team is Seed 5 in the SOUTH region?
Kentucky
What Team is Seed 6 in the SOUTH region?
Miami
What Team is Seed 7 in the SOUTH region?
Nevada
What Team is Seed 8 in the SOUTH region?
Creighton
What Team is Seed 9 in the SOUTH region?
Kansas St.
What Team is Seed 10 in the SOUTH region?
Texas
What Team is Seed 11 in the SOUTH region?
Loyola-Chi
What Team is Seed 12 in the SOUTH region?
Davidson
What Team is Seed 13 in the SOUTH region?
Buffalo
What Team is Seed 14 in the SOUTH region?
Wright St
What Team is Seed 15 in the SOUTH region?
Georgia St
What Team is Seed 16 in the SOUTH region?
UMBC
What Team is Seed 1 in the WEST region?
Xavier
What Team is Seed 2 in the WEST region?
N. Carolina
What Team is Seed 3 in the WEST region?
Michigan
What Team is Seed 4 in the WEST region?
```