APT: Tourney

Problem Statement

Class

```
public class Tourney {
   public String winner(String[] bracket, String results)
        // fill in code here
   }
}
```

In a single-elimination tournament, the schedule is determined by placing the teams in a vertical list called "the tournament bracket". In the first round the top two teams in the bracket play each other, then the next two, etc. In the second round the top two remaining teams play each other, then the next two, etc. This continues until there is only one team left -- the tournament winner.

The number of teams in the tournament should be a power of 2 -- if it isn't, byes are added to the bracket. Byes are never placed so that they will "play" each other. When a team's opponent is "bye", there is no game played and the team automatically advances as if it had won a game. Here is an example with a bracket of length 4 containing one bye (see Example 0).

```
       ROUND 1       ROUND 2
Duke
    |--------- Duke
UCLA              |
                  |-------- MIT
bye               |
    |---------- MIT
MIT
```

We have the tournament bracket (including any byes) from last year's tournament. We also have a list of the results of the games in the order in which they were played. We want to figure out who won last year's tournament. Create a class Tourney that contains a method winner that is given a `String[] bracket` and a `String results` and that returns the name of the winning team as a `String`. The bracket is given in order from top to bottom, and the results are given as a string of 'H' or 'L' indicating for each game (in the order in which the games were played) whether the Higher or Lower team from the bracket won that game. results contains a character for each real game, but not for byes.

Notes and Constraints

- It is possible that two teams with the same name could be in the tournament.
- the number of elements in `bracket` will be a power of 2 between 2 and 32 inclusive
- each element in `bracket` will have length between 1 and 50 inclusive
- each element in `bracket` will contain only uppercase characters 'A'-'Z', or will be "bye"

- no two bye elements will be placed such that they would play each other
- the length of `results` will be exactly one less than the number of non-bye elements in `bracket`
- each character of `results` will be uppercase 'H' or uppercase 'L'

Examples

1.

```
bracket = {"DUKE","UCLA","bye","MIT"}

results = "HL"

Returns:  "MIT"
```

DUKE played UCLA in the first round and MIT advanced with a bye. DUKE won since the first game was won by the team higher on the bracket. In the second round, MIT beat DUKE since the second game was won by the lower team.

2.

```
bracket = {"A","B","C","bye","D","E","F","bye"}

results = "LHHLH"

Returns "B"
```

Round 1: B beat A, D beat E
Round 2: B beat C, F beat D
Round 3: B beat F

3.

```
bracket = {"MIT","bye"}

results = ""

Returns: "MIT"
```

4.

```
bracket = {"STANFORD","bye","STANFORD","bye"}

results = "L"

Returns: "STANFORD"
```