

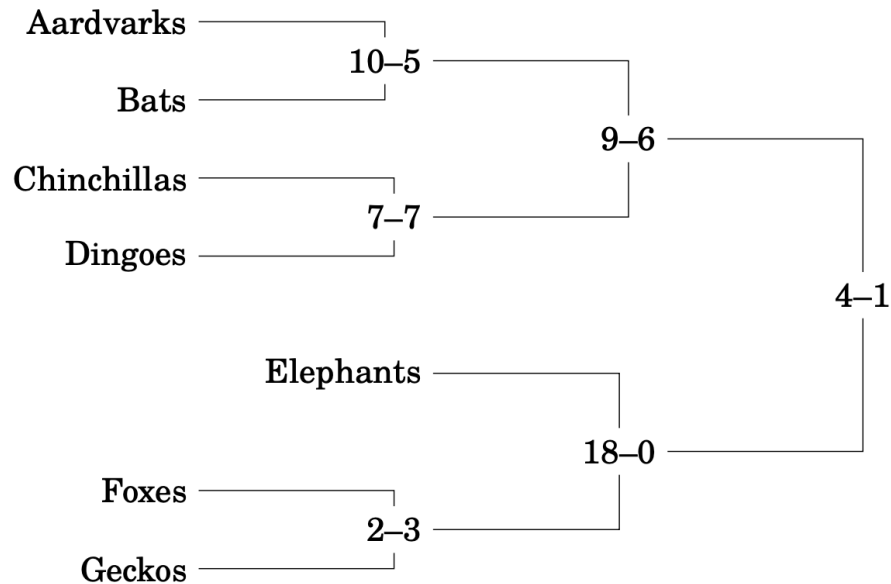
# Placement Exam

## Instructions

- Complete each task the best you can.
- You may use any real programming language of your choice, but you may not use pseudocode.
- You may use the standard library of your language.
- It's your responsibility to ensure your code is understandable. If we can't understand how your code works—even if the code may be technically correct—then we will deduct points.
- **Read the entire exam before you write anything.**

## Definitions

The Evanston Mammal League runs an annual tournament of animals. The results of last year's tournament, in which seven teams entered, can be represented visually as a single-elimination bracket with scores:



- Each leaf of the tree contains the name of a team (all of which are distinct).
- Each internal node shows the result of the match between two teams, written as *home-away*, where the team written first (higher on the page) is the “home” team and the team written second is the “away” team. (For example, in the Aardvarks–Bats match-up, the home-team Aardvarks scored 10 to the away-team Bats’ 5.)
- In case of tied scores, the home team wins.

So the above tournament is as follows.

1. Round 1:
  - Aardvarks defeat Bats, 10–5
  - Chinchillas defeat Dingoes, tied at 7 but Chinchillas are home
  - Geckos defeat Foxes, 3–2
2. Round 2:
  - Aardvarks defeat Chinchillas, 9–6
  - Elephants defeat Geckos, 18–0
3. Round 3:
  - Aardvarks defeat Elephants, 4–1

## Tasks

1. Design a data structure, class, or data type in the language of your choice for representing a single-elimination tournament with scores, as described on the previous page. Don’t include any functions or methods in your design

at this point, because we just want to understand how the information is represented. (In some languages you'll have to write a constructor.)

***Hint:* Read the rest of the exam before writing anything here.**

2. Demonstrate how to represent the example bracket from the definitions using your data structure from the previous task. In particular, show how to define a value or instance representing that example.
3. Write a method or function on brackets that counts the number rounds required to play the bracket.

For example, given the example bracket in the diagram above, the result should be 3.

4. Write a method or function on brackets that determines the winner of the given bracket.

For example, given the example bracket in the diagram above, the result should be Aardvarks.