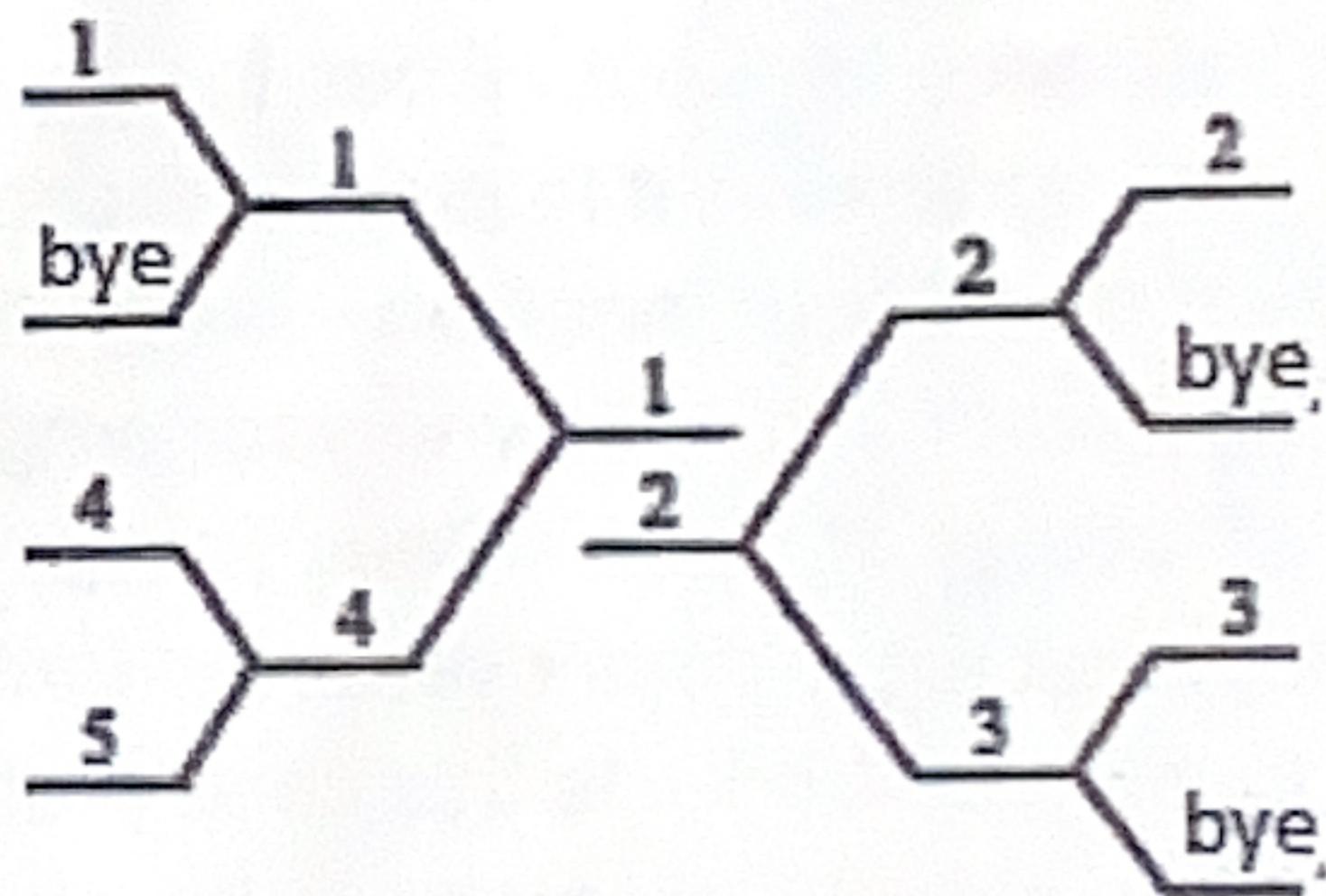
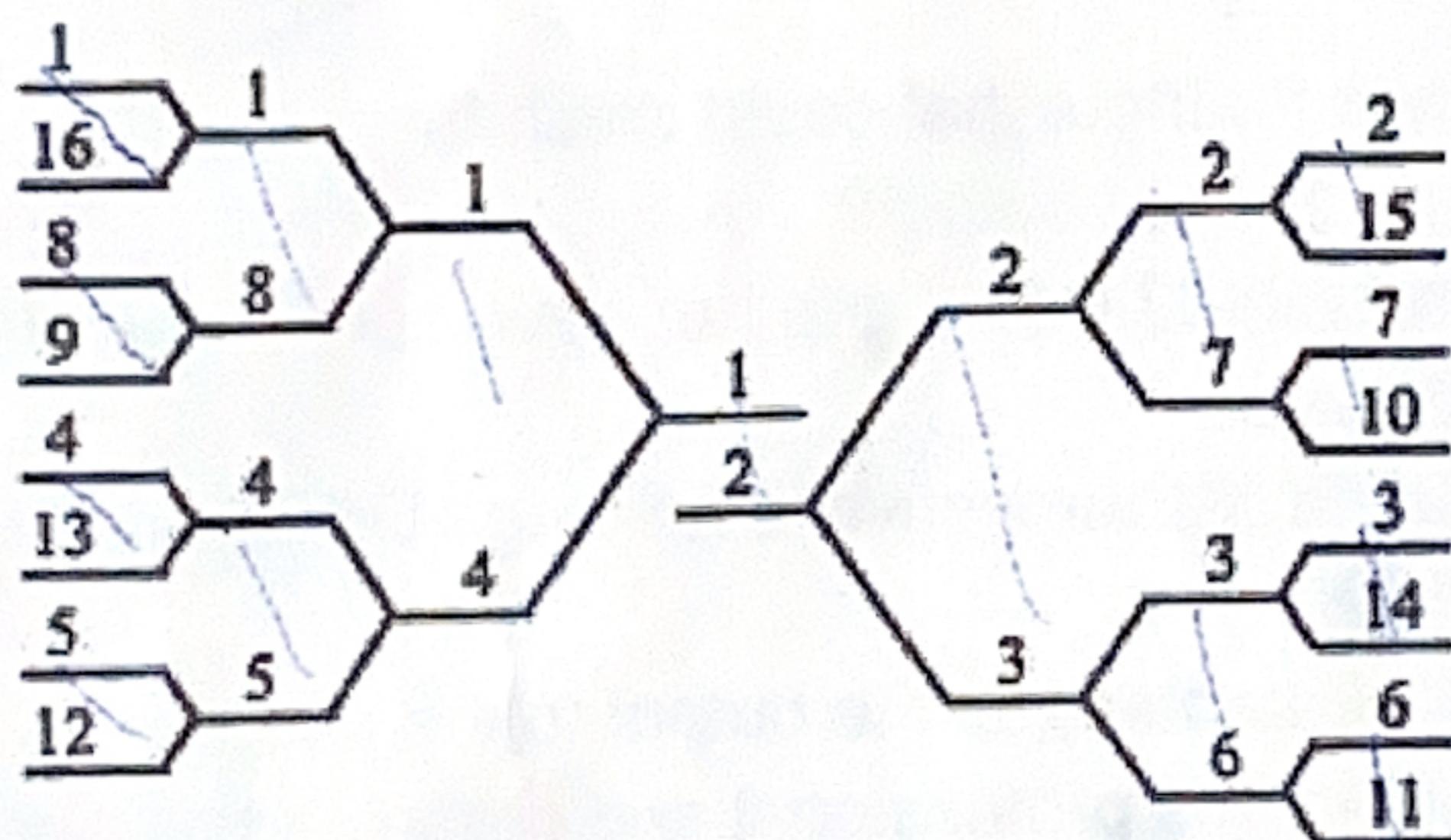


SkillsUSA Texas State 2023 Computer Programming Test

Red Rover Association: The local Red Rover Association wants a program to create and manage their monthly single elimination, seeded tournament. First, the tournament director will have a .csv file with all the teams prepared prior to setting up the tournament. We have given you a copy of this file for your convenience. You are not expected to modify the .csv file, and all .csv files used to test the program will be formatted in a similar manner. When the program runs, it should automatically load in the file of the team information. Teams are seeded by ranking with a value of 1 as the best team in the document. The file will have a minimum of 4 teams but may contain up to 25 teams. Byes should be applied only to the first round for teams of the highest seeding.

Then the program will proceed with the following 2 tasks:

1. Create a text file that outlines the tournament bracket named *bracket.txt*. This should be formatted so the tournament director can print the document and hang it for all the competitors to know who their next match will be. Given that teams are seeded, each run of the program with the same input file should create a consistent bracket. An **example** of what a seeded bracket would look like for 16 teams is found below, but you are not required to create the exact image for your tournament. We encourage you to be creative.



16 team bracket vs. 5 team bracket

2. Manage the tournament asking for the winning (or losing) team of each game. Based on the winners, it will create the next game in the tournament until the winner of the tournament is declared. Program users should not be expected to type the full name of the winning team as they work. Some system of shortening names or asking for team numbers will result in a better overall score. The goal of the program should be to make directing a tournament easier, not require them to memorize commands and team numbers.

User Stories:

As a user, I want to print a copy of the bracket for display.
As a user, I want to be able to declare the winner of a match.
As a user, I want to be able to determine the participants of the next match in the tournament.
As a user, I want to load teams in with a .csv file.
As a user, I want the bracket to be presented in standard seeding format (Team seeded 1 plays team seeded 16, 2 plays 15, etc.)

Stretch Goals:

As a user, I want to interact with the bracket using a graphical user interface.
As a user, I want to be able to print the bracket directly to a printer.
As a user, I want to interact with a graphical user interface to generate the initial bracket.

When you are done with both programs, submit your code on a USB flash drive to the test proctor. Both programs are graded with the following attributes in mind:

- Completeness - Does the program run an entire use case? Does it generally accomplish the task assigned?
- Correctness of Output - Does the program produce the output requested in the format requested?
- Validation of Input - Does the program check for input that could crash the program/cause it to provide incorrect or unexpected output?
- Internal Documentation - Is commenting provided? Could another developer come along and add functionality to your program with ease?
- Efficiency of Code - Does your program waste resources while running? Do you use optimal algorithms over brute force code?
- Quality of Work - Is this code professional? Externally, is the output to the user professional? Internally, is the code presented in a professional and organized manner?