IF100 – Spring 2021-2022
Take-Home Exam 2
Due April 21st, 2022, Thursday, 23:59 (Sharp Deadline)

Introduction

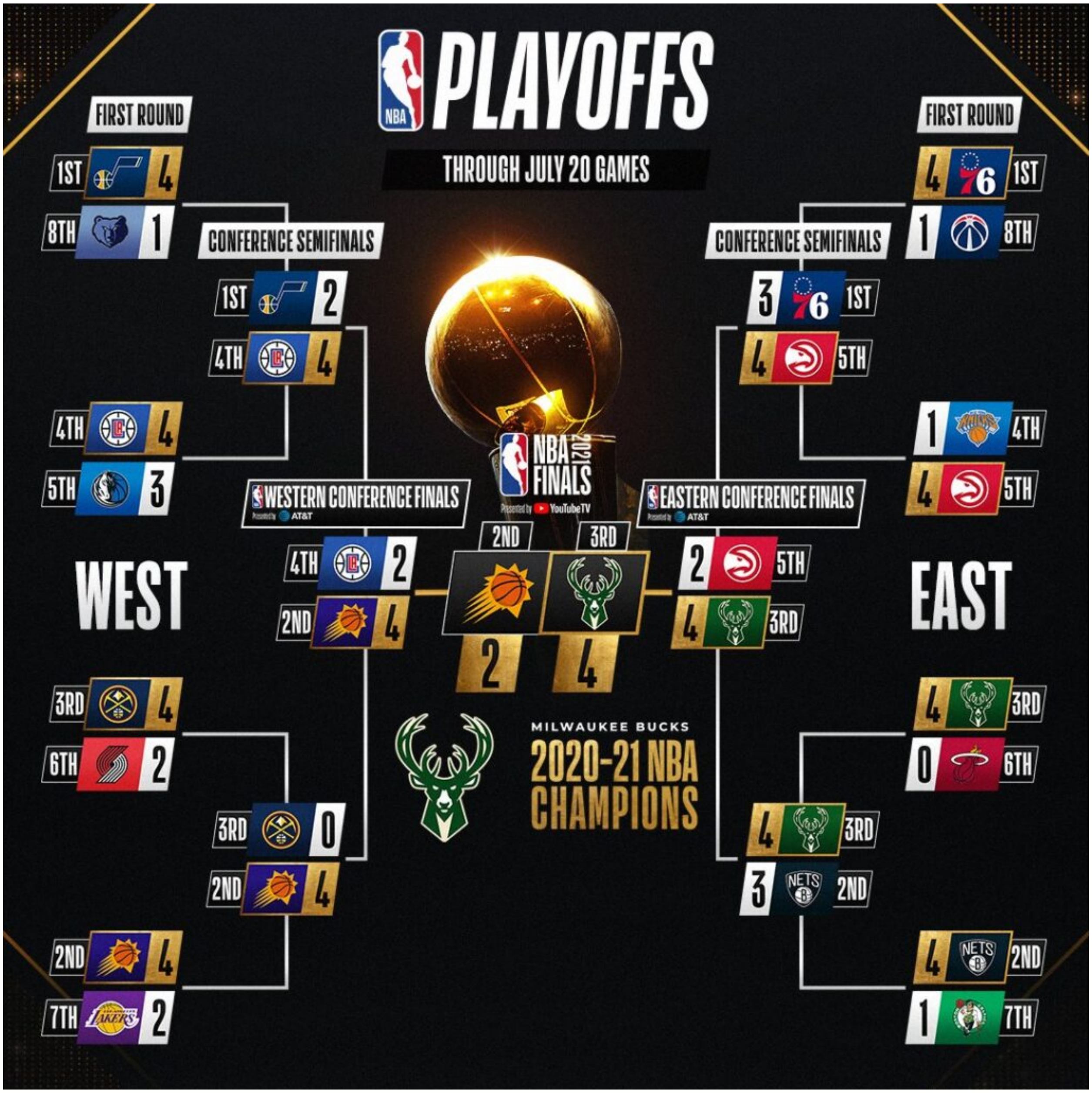
The aim of this take-home exam is to practice decision making (conditional if statements), sequences, lists and methods. The use of if statements, string methods and lists are due to the nature of the problem; that is, you cannot finish this take-home exam without using them.

Description

NBA (National Basketball Association) is a professional basketball league in North America. The current league organization divides teams into two conferences (East and West). In the regular season, teams that belong to the same conference are able to play with each other. Eight teams from each conference will participate in the playoffs. During the playoffs, matchmaking is separated by conferences once again. Teams that are in the same conference are matched in these matchmakings. There are 3 different rounds. All of these rounds are best-of-seven series, a series ends when one team wins four games, and that team advances to the next round.

In this task, we are asked to write a Python program that will take the second round (Conference Semifinals) results and a target team code from the user, and then print whether the target team makes it to the next round (Conference Finals) or not. If so, print the team that will be played against. If not, print the team that lost against.

Please see the image given on the next page.


PLAYOFFS
THROUGH JULY 20 GAMES
FIRST ROUND
1ST 4
8TH 1
4TH 4
5TH 3
3RD 4
6TH 2
2ND 4
7TH 2
CONFERENCE SEMIFINALS
1ST 2
4TH 4
3RD 0
2ND 4
WESTERN CONFERENCE FINALS
Presented by AT&T
4TH 2
2ND 4
WEST
NBA FINALS 2021
Presented by YouTube TV
2ND
3RD
2
4
MILWAUKEE BUCKS
2020-21 NBA CHAMPIONS
EASTERN CONFERENCE FINALS
Presented by AT&T
2 5TH
4 3RD
EAST
FIRST ROUND
4 1ST
1 8TH
1 4TH
4 5TH
4 3RD
0 6TH
4 2ND
1 7TH
CONFERENCE SEMIFINALS
3 1ST
4 5TH
4 3RD
3 2ND

Inputs

The inputs of your program and their order are explained below. It is extremely important to follow this order with the same characters since we automatically process your programs. ***Thus, your work will be graded as 0 unless the order is entirely correct***. Please see the "Sample Runs" section for some examples.

The prompts of the input statements to be used have to be exactly the same as the prompts of the "Sample Runs".

Here is the detailed information on the inputs and the input checks:

- Dataset - Team codes, match results
  Format: `team1-team2:teamscore1-teamscore2,team3-team4:teamscore3-teamscore4;team5-team6:teamscore5-teamscore6,team7-team8:teamscore7-teamscore8`
  i.e. **`"UTA-LAC:2-4,DEN-PHX:0-4;PHI-ATL:3-4,MIL-BKN:4-3"`**
    - Colon (":") is used to separate the matchup and the match results from each other.
    - Comma (",") is used to separate matches belonging to the same conference from each other.
    - Dash ("-") is used to separate team names and scores from each other.
    - Semicolon (";") is used to separate the conferences from each other.
        - Keep in mind that the teams that belong to the same conference will face each other in the next round.
    - You can assume that there will be exactly 4 matches separated 2 by 2 according to the conference.
    - You may assume that there won't be any duplicated team names within this input. You may also assume that the team codes will not contain any punctuation marks.
    - You may assume that one team will always win against the other. So, there will be no draw.
    - Team codes are case insensitive which means "python" and "PYTHON" and "PyThoN" should be treated the same in a manner.
    - Match results will be exactly in the following format : "integer number"-"integer number", and each integer number will contain only one digit.
    - You cannot make assumptions about the length of the team code or dataset.

- Target team. *Format:* `teamcode`
  i.e. **`"MIA"`**
  - Keep in mind that the length of the team code can be **anything**.

You may check *Sample Runs,* for more examples. However, please ***keep in mind*** that sample runs may not cover all possible cases mentioned in this document.

Output

If *`teamcode`* does not exist in the *database,* then your program should display an error message saying

"*teamcode does not exist!*"

Keep in mind that the validity check must be case insensitive; meaning that "python" and "PYTHON" and "PyThoN" should be treated the same in a manner.

If *`teamcode`* exists in the *database,* then your program should check if *`teamcode`* has won the match or not. If it couldn’t, then your program should display a message as below, where *`opponentteam`* is the team that stands against the *`teamcode`* in the dataset.

"*teamcode lost the game against opponentteam.*"

If *`teamcode`* has won the match, then your program should display a message as below, where *`nextteam`* is the other team that won the match from the same conference.

"*Next opponent of teamcode will be nextteam.*"

Keep in mind that *`teamcode`, `opponentteam`* *and* *`nextteam`* **must be written in capital** in every output.

You may check the "Sample Runs" section given below for some examples.

Sample Runs

Below, we provide some sample runs of the program that you will develop. The *italic* and **bold** phrases are inputs taken from the user. You have to display the required information in the same order and with the same words and characters as below.

Sample Run 1

```
Please enter the Conf. Semis results: UTA-LAC:2-4,DEN-PHX:0-4;PHI-ATL:3-
4,MIL-BKN:4-3
Please enter the team that you want to check: MIA
MIA does not exist!
```

Sample Run 2

```
Please enter the Conf. Semis results: UTA-LAC:2-4,DEN-PHX:0-4;PHI-ATL:3-
4,MIL-BKN:4-3
Please enter the team that you want to check: LAC
Next opponent of LAC will be PHX.
```

Sample Run 3

```
Please enter the Conf. Semis results: UTA-LaC:2-4,dEN-phX:0-4;PHI-atl:3-
4,MiL-bKn:4-3
Please enter the team that you want to check: pHx
Next opponent of PHX will be LAC.
```

Sample Run 4

```
Please enter the Conf. Semis results: UTA-LAC:2-4,DEN-PHX:0-4;PHI-ATL:3-
4,MIL-BKN:4-3
Please enter the team that you want to check: BKN
BKN lost the game against MIL.
```

Sample Run 5

```
Please enter the Conf. Semis results: LA-RoC:4-1,DeN-lac:4-3;MIL-MIA:1-
4,BoS-TOR:4-3
Please enter the team that you want to check: MIL
MIL lost the game against MIA.
```

Sample Run 6

```
Please enter the Conf. Semis results: LA-RoC:4-1,DeN-lac:4-3;MIL-MIA:1-4,BoS-TOR:4-3
Please enter the team that you want to check: LaC
LAC lost the game against DEN.
```

What and where to submit?

You should prepare (or at least test) your program using Python 3.x.x. We will use Python 3.x.x while testing your take-home exam. Let us repeat,

- You must use Google Colab to develop your code from scratch (from beginning till the end), and then submit it **through SUCourse+ only**! Once you are done with developing your code on Google Colab, then you will copy your code to the CodeRunner to see if your program can produce the correct outputs. At the end, you will submit your code through CodeRunner (and SUCourse+). You should keep your Google Colab file until the end of the semester, we might want to look at this. If you fail to provide this Google Colab file anytime in the semester, you may not earn any credits from this Take Home Exam.

- In the CodeRunner, there are some visible and invisible (hidden) test cases. You will see your final grade (including hidden test cases) before submitting your code. Thus, it will be possible to know your THE grade before submitting your solution.

- **There is no re-submission**. You don't have to complete your task in one time, you can continue from where you left last time but you should not press submit before finalizing it. Therefore, you should make sure that it's your final solution version before you submit it.

General Take-Home Exam Rules

- Successful submission is one of the requirements of the take-home exam. If, for some reason, you cannot successfully submit your take-home exam and we cannot grade it, your grade will be 0.

- There is NO late submission. You need to submit your take-home exam before the deadline. Please be careful that SUCourse+ time and your computer time may have 1-2 minutes differences. You need to take this time difference into consideration.

- You must use Google Colab to develop your code from scratch (from beginning till the end), and then submit it **through SUCourse+ only**! Once you are done with developing your code on Google Colab, then you will copy your code to the CodeRunner to see if your program can produce the correct outputs. At the end, you will submit your code through CodeRunner (and SUCourse+). You should keep your Google Colab file until the end of the semester, we might want to look at this. If you fail to provide this Google Colab file anytime in the semester, you may not earn any credits from this Take Home Exam.

- Do NOT submit your take-home exam via email or in hardcopy! SUCourse+ is the only way that you can submit your take-home exam.
- If your code does not work because of a syntax error, then we cannot grade it; and thus, your grade will be 0.
- Please submit your **own** work only. It is really easy to find "similar" programs!
- Plagiarism will not be tolerated. Please check our plagiarism policy given in the syllabus of the course.

Good luck!
Özgün Yargı
& IF100 Instructors