



Note: Please **SUBMIT** each question individually before ending the exam to receive score.
Note: This is a monitored test.

TIME REMAINING
0:37:21[End Exam](#)

Previous	Balloon Decoration	ICC Champion trophy 2025	Yet Another Chat App	eCommerce Store Design Problem	Next >
--------------------------	--------------------	--------------------------	----------------------	--------------------------------	------------------------

ICC Champion trophy 2025

Pakistan is hosting the ICC champion trophy 2025. Top eight ICC ranked teams will participate in the contest. In total fifteen matches would be played in different venues of the country. Tens of millions of people will follow it online.

To make it interesting for the fans around the globe, Pakistan cricket board has come up with different ideas. One of them is to release the win predictability of each team against all other participating teams a week in advance of the tournament. They have score cards for all the matches played among the participating teams in history. All that data is available through the [API](#). It returns a list of matches, each match contains the score cards of both of the teams. Here is the structure of the response

```
[  
  {  
    "date": "02/02/2022",  
    "team1": "Pakistan",  
    "team2": "India",  
    "venue": "MCG",  
    "scoreCardTeam1": [  
      {  
        "name": "Fakhar Zaman",  
        "score": 53  
      },  
      ...  
    ],  
    "scoreCardTeam2": [  
      {  
        "name": "Virat",  
        "score": 10  
      },  
      ...  
    ],  
    ...  
  ]
```

You have been hired by the board to build the very first version of this predictor. Your task is to write a program that will take two countries as input, say A & B and will predict the winning chances of each of these countries out of hundred by following the given criteria

1. Percentage of number of times team A has scored more than team B will represent the winning chances of A, rounded by two decimal places
2. Percentage of number of times team B has scored more than team A will represent the winning chances of B, rounded by two decimal places
3. Each match has a weightage based upon how far in the past it was played
 1. Matches played in the current year has a weightage of 10
 2. Weightage reduces by 1 with every year in the past
 3. Weightage for all the matches played before 10 or more years would be 0.5

General Example:

Let's say Pakistan and Australia have played 10 matches in the history ever, five of them were played in 2024 and 5 were played in 2022. Pakistan scored more than Australia in two matches of 2024 and three matches of 2022. The weighted percentage of the number of times Pakistan won, will represent the winning chances of Pakistan, that is 48.89% in this example. Similarly the weighted percentage of the number of times Australia won, will represent the winning chances of Australia.

Input

The input will be read from a file, it will contain names of two teams, separated by comma for which prediction is required to be made.

Output

The output would be the win prediction for the given teams in same order, separated by comma

NOTES:

1. Current year is 2024
2. Each of the teams has played minimum of 10 matches with every other team
3. Team's score is equivalent to the sum of the each player's score available in the scorecard
4. Skip the draws(matches where scores of both teams were equal)
5. Country spellings in the input would exactly match with that in response of API

Sample 1

Input:

NewZealand,Pakistan

Output:

34.72%,65.28%

Sample 2

Input:

SriLanka,Netherlands

Output:

54.95%,45.05%

Sample 3

Input:

Pakistan,Netherlands

Output:

50.61%,49.39%

Input file reading Instructions:

The input is read from a file. The filename/path of the file is passed to your program as the first command-line argument. There is no fixed name for the input file. Do not hardcode the input file name.

✓ COMPLETE

▼ Your Response

Status

Your response

```
import requests
from collections import defaultdict
import sys

def get_year_from_date(date: str) -> int:
    """
    returns year part from a date in mm/dd/yyyy format
    """
    return int(date.split("/")[-1])

# weightage of the won match depends upon the year in which it was played
def get_weightage(current_year: int, match_year: int) -> float:
    """
    returns weightage of match played at specific year
    """
    weightage = 10 - (current_year - match_year)
    if weightage > 0:
        return weightage
    else:
        return 0.5

def get_total_from_scorecard(scorecard: list) -> int:
    """
    gets total team score form provided scorecard
    """
    total = 0
    for score in scorecard:
        total += score["score"]
    return total

# winning team is the one who scored highest
def get_winning_team(match: dict) -> str:
    """
    gets winning team in the match
    returns None in case of draw
    """
    team1 = match["team1"]
    team2 = match["team2"]
    team1_score = get_total_from_scorecard(match["scoreCardTeam1"])
    team2_score = get_total_from_scorecard(match["scoreCardTeam2"])
    if team1_score > team2_score:
        winning_team = team1
    elif team1_score < team2_score:
        winning_team = team2
    else: # In case of draw
        winning_team = None
    return winning_team

def get_win_probability(matches: list, current_year: int, teams: list) -> dict:
    """
    returns win probability of both teams given in the teams list
    return value is a dict with keys as team names and values as win probability
    """
    obtained_weighted_points = defaultdict(lambda: 0)
    total_weighted_points = 0
    for match in matches:
```

```

team1 = match["team1"]
team2 = match["team2"]
# Both input teams should be in the match
# Continue otherwise
if team1 not in teams or team2 not in teams:
    continue
winning_team = get_winning_team(match)
if winning_team is not None:
    match_year = get_year_from_date(match["date"])
    match_weightage = get_weightage(current_year, match_year)
    obtained_weighted_points[winning_team] += match_weightage
    total_weighted_points += match_weightage
# Win probability from above calculations will be
win_probability = {}
for team, points in obtained_weighted_points.items():
    win_probability[team] = (points / total_weighted_points) * 100
return win_probability

if __name__ == "__main__":
    with open(sys.argv[1], "r") as file:
        teams = file.readline().strip().split(",")
    API_URL = (
        "https://1016pp2i0k.execute-api.eu-north-1.amazonaws.com/default/icc_matches"
    )
    CURRENT_YEAR = 2024
    matches = requests.get(API_URL).json()
    win_probabilites = get_win_probability(matches, CURRENT_YEAR, teams)
    team1 = teams[0]
    team2 = teams[1]
    print(f"(win_probabilites[team1]:.2f}%,", end="")
    print(f"(win_probabilites[team2]:.2f}%")

```

Test Case Result Breakdown

Test Cases Result: 3 / 3

Test Input	Your Output	Expected Output
SriLanka,Netherlands	54.95%,45.05%	54.95%,45.05%
Pakistan,Netherlands	50.61%,49.39%	50.61%,49.39%
NewZealand,Pakistan	34.72%,65.28%	34.72%,65.28%

◀ Previous Next ▶