

## **PRACTICAL: 4**

We represent scores of batsmen across a sequence of matches in a two level dictionary as follows:

```
{'match1':{'player1':57, 'player2':38}, 'match2':{'player3':9, 'player1':42}, 'match3':{'player2':41, 'player4':63, 'player3':91}}
```

Each match is identified by a string, as is each player. The scores are all integers. The names associated with the matches are not fixed (here they are 'match1', 'match2', 'match3'), nor are the names of the players. A player need not have a score recorded in all matches.

Define a Python function `orangecap(d)` that reads a dictionary `d` of this form and identifies the player with the highest total score. Your function should return a pair (`playername`, `topscore`) where `playername` is a string, the name of the player with the highest score, and `topscore` is an integer, the total score of `playername`.

The input will be such that there are never any ties for highest total score

➤ **CODE:**

```
games = {'match1':{'player1':57, 'player2':38},  
         'match2':{'player3':9, 'player1':42},  
         'match3':{'player2':41, 'player4':63, 'player3':91}}
```

```
def orangecap (d):  
    totalscore = dict()  
    for match in d:  
        for plr in d[match]:  
            if plr in totalscore:  
                totalscore[plr] += d[match][plr]  
            else:  
                totalscore[plr] = d[match][plr]  
    best = max(totalscore, key=totalscore.get)  
    return best, totalscore[best]
```

```
print(orangecap(games))
```

➤ **OUTPUT:**

```
('player3', 100)
```