Dictionary inside a list , or a list inside a dictionary is useful.

```
'''
Quick Introduction to Dictionary
'''
dict = {"SG":32, "KL":34, "BK":35, "TK":28}
print(dict)
print(dict["BK"])
dict["SY"]= 100
print(dict)
dict["SG"]= 132
print(dict["SG"])

dict2 = {101:1000, 201:340, 23:123}
print(dict2[101])   # Crash if key does not exist
#print(dict2[999])
val = dict2.get(999) # return none if value does not exist
if val == None:
    print("Does not exist")

print(val)

if 999 in dict2:
    print("Exist")
else:
    print("Not in dictionary")

val = dict2.get(999, -1) # return -1 if key does not exist

# Get all the keys
for val in dict.keys():
    print(val)
# Get all the values
for val in dict.values():
    print(val)

# Get both key and value, one pair at a time
for val in dict.items():
    print(val)

# Another way to get both key and value pair
for v1, v2 in dict.items():
    print(v1, v2)

# Remove key-value pair from dictionary
del dict["KL"] # Remove key = "KL"
print(dict)

dict.popitem() # Remove last key-value pair

# Dictionary can contain anything as value (list, number etc)
student = {}
student ["S1"] = {"name": "Alice", "major": "BICT", "GPA": 3.5}
student ["S2"] = {"name": "Benny", "major": "BEHAS", "GPA": 3.2}

for s in student.keys():
    print(s, student[s])
```

```
{'SG': 32, 'KL': 34, 'BK': 35, 'TK': 28}
35
{'SG': 32, 'KL': 34, 'BK': 35, 'TK': 28, 'SY': 100}
132
1000
Does not exist
None
Not in dictionary
SG
KL
BK
TK
SY
132
34
35
28
100
('SG', 132)
('KL', 34)
('BK', 35)
('TK', 28)
('SY', 100)
SG 132
KL 34
BK 35
TK 28
SY 100
{'SG': 132, 'BK': 35, 'TK': 28, 'SY': 100}
S1 {'name': 'Alice', 'major': 'BICT', 'GPA': 3.5}
S2 {'name': 'Benny', 'major': 'BEHAS', 'GPA': 3.2}
```

```
'''
Example of dictionary containing list
{"team A": [3, 1, 1, 5, 1, 10], "team B": [3, 1, 1, 5, 1, 10]...."}

A, 3, 1, 1, 5, 1, 10 ---> split(",")
values = ["A", "3", "1", "1", "5", "1", "10"]
key --> "A"
value --> [3, 1, 1, 5, 1, 10]
team["A"] = [...]
'''
def getTeamData():
    teams = {}
    datafile = open("TeamOne.txt", "r")
    for eachLine in datafile:
        values = eachLine.split(",")
        name = values[0]
        stat = []
        for num in values[1:]:
            stat.append(int(num))

        teams[name] = stat

    datafile.close()
    return teams

### main starts here
```

```
sgTeam = getTeamData()
print(sgTeam)
```

TeamOne.txt

```
{'A': [3, 1, 1, 5, 1, 10], 'B': [3, 1, 1, 5, 1, 10], 'c': [3, 1, 1, 5, 1, 10]}
```

```
'''
Example of dictionary containing list
{"team A": [3, 1, 1, 5, 1, 10], "team B": [3, 1, 1, 5, 1, 10]...."}

A, 3, 1, 1, 5, 1, 10 ---> split(",")
values = ["A", "3", "1", "1", "5", "1", "10"]
key --> "A"
value --> [3, 1, 1, 5, 1, 10]
team["A"] = [...]
'''
def getTeamData():
    teams = {}
    datafile = open("TeamOne.txt", "r")
    for eachLine in datafile:
        values = eachLine.split(",")
        name = values[0]
        stat = []
        for num in values[1:]:
            stat.append(int(num))

        teams[name] = stat

    datafile.close()
    return teams

def updateTeam(teamStat, gScored, gAllowed):
    if gScored > gAllowed:
        teamStat[0] += 1
        teamStat[5] += 3
    elif gScored < gAllowed:
        teamStat[1] += 1
    else:
        teamStat[2] += 1
        teamStat[5] += 1
    teamStat[3] += gScored
    teamStat[4] += gAllowed

### main starts here
sgTeams = getTeamData()

# Loop to allow user to enter match result and update the dictonary
matchResult = input("Enter name, goals scored, goals allowed: ")
while matchResult != "":
    name, gScored, gAllowed = matchResult.split() # "A 5 2"
    gScored, gAllowed = int(gScored), int(gAllowed) # Convert string to int
    teamStat = sgTeams[name]
    updateTeam(teamStat, gScored, gAllowed)
```

```
    matchResult = input("Enter name, goals scored, goals allowed")

# Show result
for team in sgTeams:
    print(team, sgTeams[team])


Enter name, goals scored, goals allowed: B 5 2
Enter name, goals scored, goals allowedA 2 1
Enter name, goals scored, goals allowed
A [4, 1, 1, 7, 2, 13]
B [4, 1, 1, 10, 3, 13]
c [3, 1, 1, 5, 1, 10]



'''
Finding Max
'''
myList = [["a", 10, 10], ["x", 40, 200], ["m", 20, 5], ["W", 50, 40]]

index = 0
highest = myList[0][2]

for idx in range(1, len(myList)):
    if highest < myList[idx][2]:
        index = idx
        highest = myList[idx][2]

print(index)
    print(highest)




import random
def getDummyData(students):
    for idx in range(1, 11):
        name = "S{}".format(idx)
        score = random.randint(0, 100)
        students[name] = score

def showMenu():
    print("1 Add")
    print("2 Update")
    print("3 Remove")
    print("4 Retrieve score by name")
    print("5 Retrieve name(s) by score")
    print("0 Quit")

def main():
    students = {}
    getDummyData(students)  # Call function to fill up dictionary
    showMenu()
    option = int(input("Enter an option: "))
    while option !=0:
        if option == 1:
            name = input("Enter student name: ")
            score = int(input("Enter score:  "))
```

```
            students[name] = score
        elif option == 2:
            name = input("Enter student name: ")
            score = int(input("Enter score:  "))
            if name in students:
                students[name] = score
            else:
                print("No such student")
            students[name] = score
        elif option == 3:
            name = input("Enter student name: ")
            if name in students:
                students.pop(name)
            else:
                print("No such student")
        elif option == 4:       # Retrieve score by name
            name = input("Enter student name: ")
            score = students.get(name, -99)
            if score == -99:
                print("No such student")
            else:
                print("Student {} has score {}".format(name, score))
        elif option == 5:
            score = int(input("Enter score: "))
            for eachPair in students.items(): # Alternatively, can use for sName,
sScore in students.items()
                if eachPair[1] == score:        # if sScore == score:
                    print(eachPair[0])          # print(sName)
        else:
            print("Invalid option")
        showMenu()
            option = int(input("Enter a option"))
```