

# Forum:[Hack Forums](#)

Thread: [Dictionaries](#)

**twistingtide** - March 29, 2017, 11:36 pm

1. So I have a dictionary with the set up below

The actor name, the movies they were in, and number to the right of the movies is the actors popularity

```
dictionary = {1: ['Actor1', {movie1, movie3, movie7}, 89], 2:['Actor35, {movie3, movie7}, 57],  
3:['Actor7',{movie7} 64], 4:['Actor9, {movie90} 3 ]}
```

2. So If someone types in the number 3 (Or any number within the length of my dictionary they go to that actor) into my input they go to Actor7

3. Once they are at actor 7 they look at ALL the movies the actor has been in and compare to all the movies the other actors have been in

4. If any of the actors share a movie append there name to a new list in the order of decreasing popularity(Popularity is the last element in the list)

So it should look like this if they type in 3 because the only actors who share a movie with actor 7 is actor1 and actor35:

```
final_list = [Actor1,Actor35]
```

Thanks guys!!!?

**(???)?(???? - March 30, 2017, 6:25 am**

I'm on my phone, but to begin, you need to put the movies in a list instead of a dictionary (the {movies1...} part).

**hunter24957** - March 30, 2017, 7:18 am

NGL, that's the worst layout of a dictionary I've ever seen. I would fix it for you but I'm still not sure what you're going for...

Maybe use something like this:

PHP Code

```
<?php
```

```
{'Actor1': [movie1, movie3, movie7, 89], 'Actor35': [movie3, movie7, 57], 'Actor7': [movie7, 64],  
'Actor9': [movie90, 3]}
```

And your dictionary keys are redundant, you can just use a list...

**(???)?(???? - March 30, 2017, 7:57 am**

This feels like homework to me, so I'm going to make it fun! Good ol one liner :D

```
Code actors = [  
    ("Actor 1", ["Movie 1", "Movie 3", "Movie 7"], 9),
```

```

("Actor 35", ["Movie 3", "Movie 7"], 57),
("Actor 7", ["Movie 7"], 64),
("Actor 9", ["Movie 90"], 3),
]
get_actors = lambda index: list(map(lambda x: x[0], sorted(((actor[0], actor[2]) for actor in actors if
actor[0] is not actors[index - 1][0] and list(filter(lambda movie: movie in actors[index - 1][1],
actor[1]))), key=lambda x: x[1], reverse=True)))
print(get_actors(3))

```

Folks - March 30, 2017, 8:16 am

(Mar 30, 2017 07:57 AM)(?°°°)?( ???? Wrote: [CITE](#) This feels like homework to me, so I'm going to make it fun! Good ol one liner :D

```

Code actors = [
    ("Actor 1", ["Movie 1", "Movie 3", "Movie 7"], 9),
    ("Actor 35", ["Movie 3", "Movie 7"], 57),
    ("Actor 7", ["Movie 7"], 64),
    ("Actor 9", ["Movie 90"], 3),
]
get_actors = lambda index: list(map(lambda x: x[0], sorted(((actor[0], actor[2]) for actor in actors if
actor[0] is not actors[index - 1][0] and list(filter(lambda movie: movie in actors[index - 1][1],
actor[1]))), key=lambda x: x[1], reverse=True)))
print(get_actors(3))
One liners gave me an aneurism until I finally started to understand how to read through them lol

```

(?°°°)?( ???? - March 30, 2017, 8:16 am

(Mar 30, 2017 08:16 AM)Folks Wrote: [CITE](#) One liners gave me an aneurism until I finally started to understand how to read through them lol  
I wrote it out first and then I looked at it and figured I could one line it so I did.