



Arbeitsblatt 5



Schreiben Sie eine Funktion `multi_rem(li, di)`, die

- eine Liste und ein Dictionary als Argument nimmt
- für jedes Element `x` der Liste den Eintrag des Dictionary mit `x` als Schlüssel aus `di` löscht
- das Dictionary ohne die gelöschten Elemente zurückgibt
- `di.pop(k)` entfernt den Eintrag mit Schlüssel `k` aus `di` und liefert dessen Wert zurück
- `di.pop(k)` löst Fehler aus, wenn Schlüssel nicht in `di`



Beispiel für `multi_rem(li, di)`:

```
di = { 1:2, 2:2, 3:4 }  
li = [ 2, 4, 6 ]  
nd = multi_rem( li, di )  
  
nd ist  
{ 1:2, 3:4 }
```



Schreiben Sie eine Funktion

`rename_entry(di, key, new_key)`, die

- den Eintrag mit Schlüssel `key` in `di` so ändert, dass `new_key` der Schlüssel ist
- existiert der neue Schlüssel bereits, soll die Liste nicht geändert und `False` zurückgegeben werden

```
>>> rename_entry({1:2, 3:4}, 3, 5 )
```

```
{1:2, 5:4}
```



Die Funktion `di.get(k, [x])` liefert `di[k]`, wenn der Schlüssel `k` vorhanden ist, ansonsten `x`. `[x]` heißt, der Parameter ist optional, default: `None`.

Schreiben Sie eine Funktion `schlsl_oder_falsch(di, x)`, die

- ein Dictionary und eine Variable `x` als Argument nimmt,
- den Typ von `di[x]` zurückliefert, falls `x` als Schlüssel in `di` vorkommt
- `False` sonst