

1. (6 Punkte) Was erscheint auf der Konsole? Notiere die einzelnen Ausgaben aus Platzgründen horizontal mit Komma getrennt, schreibe **Leerzeile** für eine Leerzeile.

```
m = {'a': 'gb', 'b': 'g', 'g': 'b'}
print(m['a'])
print('a' in m)
print('gb' in m)
print('b' in m['g'])
print(m['b'] in m)
print(len(m))
```

**Lösung:** gb, True, False, True, True, 3

2. (6 Punkte) Was erscheint auf der Konsole? Notiere die einzelnen Ausgaben aus Platzgründen horizontal mit Komma getrennt, schreibe **Leerzeile** für eine Leerzeile.

```
m = {'a': 'd', 'b': 'a', 'g': 'abba', 'k': 'bob'}
print(len(m))
print(m['g'])
print(m['b'] in m['g'])
print('bob' in m)
print(m['b'] in m.values())
print(m['a'] in m.keys())
```

**Lösung:** 4, abba, True, False, True, False

3. (6 Punkte) G sei wie abgebildet ein dictionary, bei dem jedem key ein weiteres dictionary zugeordnet ist. Schreibe Code, der alle Werte des dictionaries ausgibt, das in G dem Schlüssel 'e' zugeordnet ist.

```
G = {
    'a': {'f': 8, 'b': 10},
    'b': {'d': 2},
    'c': {'b': 1},
    'd': {'c': -2},
    'e': {'d': -1, 'b': -4},
    'f': {'e': 1}
}
```

**Lösung:**

```
for x in G['e'].values():
    print(x)
```

4. (2 Punkte) Schreibe eine Schleife, die die keys des dictionaries ausgibt.

```
mu4 = {1:5, 2:7, 3:9, 4:9}
```

**Lösung:**

```
for x in mu4:
    print(x)
```

5. (4 Punkte) Implementiere die Funktion **gerade**

```
def gerade(g):
    """
    g: dictionary, das Buchstaben ganze Zahlen zuordnet.
    returns: sortierte Liste aller vorkommenden geraden values.

    z.B:
    >>> G = {'a': 3, 'c': 7, 'D': 10, 'e': 8, 'B': 7}
    >>> gerade(G)
    [8, 10]
    """
```

**Lösung:**

```
def gerade(g):  
    tmp = []  
    for x in g.values():  
        if x%2 == 0:  
            tmp.append(x)  
    return sorted(tmp)
```