```
Exercise in Basic Python data structures.

Reproduce the following python interactive listing using different data of
your choosing, e.g., choose different variable names and data strings other
than ‘Bob Smith’ and ‘Sue Jones’

>>> bob = ['Bob Smith', 42, 30000, 'software']
>>> sue = ['Sue Jones', 45, 40000, 'hardware']

>>> bob[0], sue[2]             # fetch name, pay
('Bob Smith', 40000)

>>> bob[0].split()[-1]         # what's bob's last name?
'Smith'

>>> sue[2] *= 1.25             # give sue a 25% raise
>>> sue

['Sue Jones', 45, 50000.0, 'hardware']

>>> people = [bob, sue ]        # reference in list of lists

>>> for person in people:
        print(person)

['Bob Smith', 42, 30000, 'software']
['Sue Jones', 45, 50000.0, 'hardware']

>>> people[1][0]

'Sue Jones'

>>> for person in people:
        print(person[0].split()[-1])          # print last names
        person[2] *= 1.20                     # give each a 20% raise

Smith
Jones

>>> for person in people: print(person[2])     # check new pay

36000.0
60000.0

>>> pays = [person[2] for person in people]    # collect all pay
>>> pays

[36000.0, 60000.0]

>>> pays = map((lambda x: x[2]), people)   # ditto (map is a generator in
3.X)

>>> list(pays)

[36000.0, 60000.0]
```

```
>>> sum(person[2] for person in people)       # generator expression, sum
built-in

96000.0

>>> people.append(['Tom', 50, 0, None])
>>> len(people)

3

>>> people[-1][0]

'Tom'

>>> NAME, AGE, PAY = range(3)                # 0, 1, and 2
>>> bob = ['Bob Smith', 42, 10000]
>>> bob[NAME]

'Bob Smith'

>>> PAY, bob[PAY]

(2, 10000)

>>> bob = [['name', 'Bob Smith'], ['age', 42], ['pay', 10000]]
>>> sue = [['name', 'Sue Jones'], ['age', 45], ['pay', 20000]]
>>> people = [bob, sue]

>>> for person in people:
        print(person[0][1], person[2][1])     # name, pay

Bob Smith 10000
Sue Jones 20000

>>> [person[0][1] for person in people]       # collect names

['Bob Smith', 'Sue Jones']

>>> for person in people:
        print(person[0][1].split()[-1])       # get last names
        person[2][1] *= 1.10                  # give a 10% raise

Smith
Jones

>>> for person in people: print(person[2])

['pay', 11000.0]
['pay', 22000.0]

>>> for person in people:
        for (name, value) in person:
            if name == 'name': print(value)   # find a specific field

Bob Smith
```

```
Sue Jones

>>> def field(record, label):
        for (fname, fvalue) in record:
            if fname == label:              # find any field by name
                return fvalue

>>> field(bob, 'name')
'Bob Smith'

>>> field(sue, 'pay')
22000.0

>>> for rec in people:
        print(field(rec, 'age'))              # print all ages

42
45

>>> bob = {'name': 'Bob Smith', 'age': 42, 'pay': 30000, 'job': 'dev'}
>>> sue = {'name': 'Sue Jones', 'age': 45, 'pay': 40000, 'job': 'hdw'}


>>> bob['name'], sue['pay']            # not bob[0], sue[2]

('Bob Smith', 40000)

>>> bob['name'].split()[-1]

'Smith'

>>> sue['pay'] *= 1.10
>>> sue['pay']

44000.0

>>> bob = dict(name='Bob Smith', age=42, pay=30000, job='dev')
>>> sue = dict(name='Sue Jones', age=45, pay=40000, job='hdw')


>>> bob

{'pay': 30000, 'job': 'dev', 'age': 42, 'name': 'Bob Smith'}

>>> sue

{'pay': 40000, 'job': 'hdw', 'age': 45, 'name': 'Sue Jones'}

>>> sue = {}
>>> sue['name'] = 'Sue Jones'
>>> sue['age']  = 45
>>> sue['pay']  = 40000
>>> sue['job']  = 'hdw'
>>> sue

```

```
{'job': 'hdw', 'pay': 40000, 'age': 45, 'name': 'Sue Jones'}


>>> names  = ['name', 'age', 'pay', 'job']
>>> values = ['Sue Jones', 45, 40000, 'hdw']
>>> list(zip(names, values))

[('name', 'Sue Jones'), ('age', 45), ('pay', 40000), ('job', 'hdw')]

>>> sue = dict(zip(names, values))
>>> sue

{'job': 'hdw', 'pay': 40000, 'age': 45, 'name': 'Sue Jones'}

>>> fields = ('name', 'age', 'job', 'pay')
>>> record = dict.fromkeys(fields, '?')
>>> record

{'job': '?', 'pay': '?', 'age': '?', 'name': '?'}

>>> bob

{'pay': 30000, 'job': 'dev', 'age': 42, 'name': 'Bob Smith'}

>>> sue

{'job': 'hdw', 'pay': 40000, 'age': 45, 'name': 'Sue Jones'}

>>> people = [bob, sue]             # reference in a list
>>> for person in people:
        print(person['name'], person['pay'], sep=', ')    # all pay

Bob Smith, 30000
Sue Jones, 40000

>>> for person in people:
        if person['name'] == 'Sue Jones':   # fetch sue's pay
            print(person['pay'])

40000

>>> names = [person['name'] for person in people]         # collect names
>>> names

['Bob Smith', 'Sue Jones']

>>> list(map((lambda x: x['name']), people))              # ditto, generate

['Bob Smith', 'Sue Jones']

>>> sum(person['pay'] for person in people)               # sum all pay

70000

>>> for person in people:
```

```
        print(person['name'].split()[-1])               # last name
        person['pay'] *= 1.10                           # a 10% raise

Smith
Jones

>>> for person in people: print(person['pay'])

33000.0
44000.0

>>> bob2 = {'name': {'first': 'Bob', 'last': 'Smith'},
            'age':  42,
            'job':  ['software', 'writing'],
            'pay':  (40000, 50000)}

>>> bob2['name']                            # bob's full name

{'last': 'Smith', 'first': 'Bob'}

>>> bob2['name']['last']                    # bob's last name\

'Smith'

>>> bob2['pay'][1]                          # bob's upper pay

50000

>>> for job in bob2['job']: print(job)      # all of bob's jobs

software
writing

>> bob2['job'][-1]                          # bob's last job

'writing'

>>> bob2['job'].append('janitor')           # bob gets a new job

>>> bob2

{'job': ['software', 'writing', 'janitor'], 'pay': (40000, 50000), 'age': 42,
'name':

{'last': 'Smith', 'first': 'Bob'}}

>>> bob = dict(name='Bob Smith', age=42, pay=30000, job='dev')
>>> sue = dict(name='Sue Jones', age=45, pay=40000, job='hdw')
>>> bob

{'pay': 30000, 'job': 'dev', 'age': 42, 'name': 'Bob Smith'}

>>> db = {}
>>> db['bob'] = bob                      # reference in a dict of dicts
>>> db['sue'] = sue
```

```
>>> db['bob']['name']                   # fetch bob's name

'Bob Smith'

>>> db['sue']['pay'] = 50000            # change sue's pay
>>> db['sue']['pay']                    # fetch sue's pay

50000

>>> db

{'bob': {'pay': 30000, 'job': 'dev', 'age': 42, 'name': 'Bob Smith'}, 'sue':
{'pay': 50000, 'job': 'hdw', 'age': 45, 'name': 'Sue Jones'}}

>>> import pprint
>>> pprint.pprint(db)

{'bob': {'age': 42, 'job': 'dev', 'name': 'Bob Smith', 'pay': 30000},
 'sue': {'age': 45, 'job': 'hdw', 'name': 'Sue Jones', 'pay': 50000}}


>>> for key in db:
        print(key, '=>', db[key]['name'])

bob => Bob Smith
sue => Sue Jones

>>> for key in db:
        print(key, '=>', db[key]['pay'])

bob => 30000
sue => 50000

>>> for key in db:
        print(db[key]['name'].split()[-1])
        db[key]['pay'] *= 1.10

Smith
Jones

>>> for record in db.values(): print(record['pay'])

33000.0
55000.0

>>> x = [db[key]['name'] for key in db]
>>> x

['Bob Smith', 'Sue Jones']

>>> x = [rec['name'] for rec in db.values()]
>>> x

['Bob Smith', 'Sue Jones']
```

```
>>> db['tom'] = dict(name='Tom', age=50, job=None, pay=0)
>>> db['tom']

{'pay': 0, 'job': None, 'age': 50, 'name': 'To'}

>>> db['tom']['name']

'Tom'

>>> list(db.keys())

['bob', 'sue', 'tom']

>>> len(db)

3

>>> [rec['age'] for rec in db.values()

[42, 45, 50]
```