CS235 HW2 – Teng Xu

1) (7 pts) Write a function that takes as parameters the set of skills required in a job ad, the set of your skills, and returns the following:
   a. whether you have all the skills required for a job,
   b. the skills that you have (out of the ones listed in a job ad),
   c. the skills that you need to acquire for a job,
   d. your skills that would not be used in a job,
   e. how well do you fit the job (percentage of the skills that you have out of all required by a job)，
   f. how well does the job fit you (percentage of your skills required by a job),
   g. all the skills that you would have after working at this job.

Output：

```
In [9]: Myskill = {'eat','sleep','work hard','read'}

In [10]: Jobrequire = {'work hard','read','dance'}

In [11]: job(Myskill, Jobrequire)
False
{'read', 'work hard'}
{'dance'}
{'sleep', 'eat'}
0.6666666666666666
0.5
{'dance', 'read', 'sleep', 'eat', 'work hard'}
```

2) (4 pts) Define a function that takes as parameters a set of engineers and a set of projects and returns all possible matchings (engineer-project), and the number of such matchings. Define the 2 sets and test the function. Suggest a relation (matching) between these (or modified) sets that is a bijection. Explain why it is a bijection.

Output:

```
In [37]: Eng = {'a','b','c','d'}

In [38]: Project = {'project1','project2','project3','project4'}

In [39]: matching(Eng,Project)
number:  16
Out[39]:
{('a', 'project1'),
 ('a', 'project2'),
 ('a', 'project3'),
 ('a', 'project4'),
 ('b', 'project1'),
 ('b', 'project2'),
 ('b', 'project3'),
 ('b', 'project4'),
 ('c', 'project1'),
 ('c', 'project2'),
 ('c', 'project3'),
 ('c', 'project4'),
 ('d', 'project1'),
 ('d', 'project2'),
 ('d', 'project3'),
 ('d', 'project4')}
```

Relation: {(‘a’, ‘project1’), (‘b’,’project2’), (‘c’,’project3’), (‘d’,’project4’)}

This is bijection because each element of one set is paired with exactly one element of the other set, and each element of the other set is paired with exactly one element of the first set.

3) (5 pts) Let C be a set of countries and D be a relation, such that 2 countries are related if you can get by car from one to another. Define a function that outputs all the clusters - sets of countries, such that you can visit all the countries in each set in one car ride. Define another function that outputs the minimum number of flights that you need to visit all the countries and return to the starting point. Test these functions on a set C consisting of at least 5 countries and a relation D defined by you, but based on geography of the countries.

Output:

```
In [47]: D = {('Canda', 'Canda'),
    ...:  ('Canda', 'USA'),
    ...:  ('Canda', 'Mexico'),
    ...:  ('China', 'China'),
    ...:  ('China', 'Russia'),
    ...:  ('Mexico', 'Mexico'),
    ...:  ('Mexico', 'USA'),
    ...:  ('Mexico', 'Canda'),
    ...:  ('Russia', 'China'),
    ...:  ('Russia', 'Russia'),
    ...:  ('USA', 'Canda'),
    ...:  ('USA', 'Mexico'),
    ...:  ('USA', 'USA')}

In [48]: C = {"China","Russia","USA","Canda","Mexico"}

In [49]: country(C,D)
Out[49]: {frozenset({'Canda', 'Mexico', 'USA'}), frozenset({'China', 'Russia'})}

In [51]: minimumFlights(C, D)
Out[51]: 2
```

4) (4 pts) Hashing function (mapping keys into hash table locations) is defined as h(k)=k mod m, where m = 11. Two keys are equivalent if they are put into the same location. Prove that this is an equivalence relation (see exercise in the online notes). How many slots would be occupied by the keys ranging from from 1100 to 1150? List the numbers that fall in the same slots. What is the biggest number of collisions (keys belonging to the same slot)?

The biggest number of collisions is 5.
Output:

```
In [3]: hash(set(range(1100,1151)))
Out[3]:
{frozenset({1105, 1116, 1127, 1138, 1149}),
 frozenset({1109, 1120, 1131, 1142}),
 frozenset({1110, 1121, 1132, 1143}),
 frozenset({1107, 1118, 1129, 1140}),
 frozenset({1108, 1119, 1130, 1141}),
 frozenset({1102, 1113, 1124, 1135, 1146}),
 frozenset({1100, 1111, 1122, 1133, 1144}),
 frozenset({1106, 1117, 1128, 1139, 1150}),
 frozenset({1103, 1114, 1125, 1136, 1147}),
 frozenset({1104, 1115, 1126, 1137, 1148}),
 frozenset({1101, 1112, 1123, 1134, 1145})}
```