```
# Python3 code to demonstrate working of
# Join Tuples if similar initial element
# Using loop

# initializing list
test_list = [(5, 6), (5, 7), (6, 8), (6, 10), (7, 13)]

# printing original list
print("The original list is : " + str(test_list))

# Join Tuples if similar initial element
# Using loop
res = []
for sub in test_list:
    if res and res[-1][0] == sub[0]:
        res[-1].extend(sub[1:])
    else:
        res.append([ele for ele in sub])
res = list(map(tuple, res))

# printing result
print("The extracted elements : " + str(res))
```

Output:

```
The original list is : [(5, 6), (5, 7), (6, 8), (6, 10), (7, 13)]
The extracted elements : [(5, 6, 7), (6, 8, 10), (7, 13)]
```

Set

Set is unordered collection.
Set is unordered collection, where insertion order not preserved. This insertion order changes from one insertion to another.
Set does not allow duplicates elements or items.
Set is non index based collection or non sequence.
Python support two set types

1. set → Mutable
2. frozenset → Immutable

A set object is an unordered collection of distinct hashable objects. Common uses include membership testing, removing duplicates from a sequence, and computing mathematical operations such as intersection, union, difference, and symmetric difference.

Set uses **hashing data structure** for organizing objects.

Q: What is hash value?
Hash value is an integer value, which is used in hash based structures for finding key.

Q:How to find hash value of object?
hash() function of python returns hash value of object

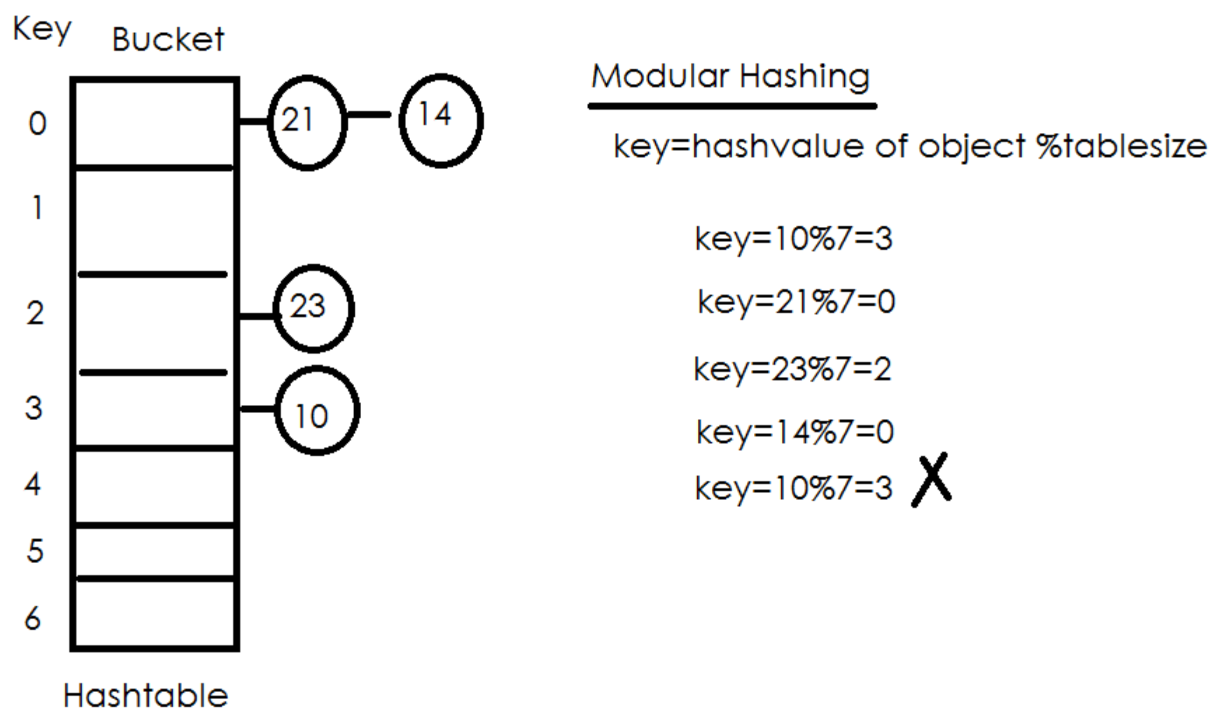
All immutable objects are hashable.

According to object rule, if two objects are equal, it must generate same hash value.

```
>>> a=10
>>> hash(a)
10
>>> b=20
>>> hash(b)
20
>>> a==b
False
>>> c=10
>>> a==c
True
>>> hash(c)
10
>>> str1="abc"
>>> hash(str1)
3733170137161483824
>>> str2="abc"
>>> hash(str2)
3733170137161483824
>>> str1==str2
True
```

```
>>> f1=1.5
>>> f2=1.5
>>> f1==f2
True
>>> hash(f1)
1152921504606846977
>>> hash(f2)
1152921504606846977
>>> id(f1)
125860503888
>>> t1=(10,20)
>>> hash(t1)
-4873088377451060145
>>> l1=[10,20]
>>> hash(l1)
Traceback (most recent call last):
  File "<pyshell#26>", line 1, in <module>
    hash(l1)
TypeError: unhashable type: 'list'
```

According to hashing, there is hashtable. Each location in hashtable is identified with a unique number called key. This key is generated based on hashing alg.



How to create set?

1. Use a comma-separated list of elements within braces: {'jack', 'sjoerd'}
2. Use a set comprehension: {c for c in 'abracadabra' if c not in 'abc'}
3. Use the type constructor: set(), set('foobar'), set(['a', 'b', 'foo'])

Example:

```
>>> set1={}
>>> type(set1)
<class 'dict'>
```