In [1]:

```
d1={}
i=True
def add():
    n=input('enter contact name')
    p=int(input('enter contact number'))
    d1.update({n:p})
    print('contact added sucessfully!')
    print(d1)
def update():
    a=input('enter contact name to update')
    if(a in d1):

        b=int(input('enter contact number'))
        d1.update({a:b})
    else:
        print('contact not found')
def search():
    c=input('enter contact name to search')
    if(c in d1):
        print(d1.get(c))
    else:
        print('contact not found')
def delete():
    d=input('enter contact to delete')
    if(d in d1):
        d1.pop(d)
        print('contact deleted sucessfully')




while(i):
    print('enter any of the option')
    print('1.add 2.update 3.search  5.delete 6.display 0.exit')
    i=int(input())
    if(i==1):
        add()
    if(i==2):
        update()
    if(i==3):
        search()

    if(i==5):
        delete()
    if(i==6):
        print(d1)
    if(i==0):
        print('exited from phonebook')

```

```
enter any of the option
1.add 2.update 3.search  5.delete 6.display 0.exit
1
enter contact nameahad
enter contact number8585
contact added sucessfully!
```

```
{'ahad': 8585}
enter any of the option
1.add 2.update 3.search  5.delete 6.display 0.exit
2
enter contact name to updateahad
enter contact number2525
enter any of the option
1.add 2.update 3.search  5.delete 6.display 0.exit
6
{'ahad': 2525}
enter any of the option
1.add 2.update 3.search  5.delete 6.display 0.exit
1
enter contact namesamad
enter contact number2525
contact added sucessfully!
{'ahad': 2525, 'samad': 2525}
enter any of the option
1.add 2.update 3.search  5.delete 6.display 0.exit
2
enter contact name to updateajay
contact not found
enter any of the option
1.add 2.update 3.search  5.delete 6.display 0.exit
6
{'ahad': 2525, 'samad': 2525}
enter any of the option
1.add 2.update 3.search  5.delete 6.display 0.exit
0
exited from phonebook
```

In [4]:
```
.d={'a':'b','c':'d'}
d.get('a')
```

Out[4]: 'b'

# SET

In [ ]:
```
set is a collection which is unordered and unindexed

```

In [1]:
```
s={}
type(s)
```

Out[1]: dict

In [3]:
```
s={4,5,3}
type(s)
```

Out[3]: set

In [14]:
```
f={1,2,2,2,3,1,1,3,4,4,5}
print(f)
print(min(f))
print(max(f))
print(len(f))
print(sum(f))
print(sorted(f))
```

```
{1, 2, 3, 4, 5}
1
5
5
15
[1, 2, 3, 4, 5]
```

In [15]:
```
print(dir(set))
```

```
['__and__', '__class__', '__contains__', '__delattr__', '__dir__', '__doc__',
'__eq__', '__format__', '__ge__', '__getattribute__', '__gt__', '__hash__', '__
iand__', '__init__', '__init_subclass__', '__ior__', '__isub__', '__iter__', '_
_ixor__', '__le__', '__len__', '__lt__', '__ne__', '__new__', '__or__', '__rand
__', '__reduce__', '__reduce_ex__', '__repr__', '__ror__', '__rsub__', '__rxor_
_', '__setattr__', '__sizeof__', '__str__', '__sub__', '__subclasshook__', '__x
or__', 'add', 'clear', 'copy', 'difference', 'difference_update', 'discard', 'i
ntersection', 'intersection_update', 'isdisjoint', 'issubset', 'issuperset', 'p
op', 'remove', 'symmetric_difference', 'symmetric_difference_update', 'union',
'update']
```

In [16]:
```
f.add('5')
```

In [19]:
```
f.add(7)
print(f)
```

```
{1, 2, 3, 4, 5, '5', 67, 7, '56'}
```

In [20]:
```
f.add('678')
print(f)
```

```
{1, 2, 3, 4, 5, '5', 67, 7, '56', '678'}
```

In [21]:
```
f.add(0)
print(f)
```

```
{0, 1, 2, 3, 4, 5, '5', 67, 7, '56', '678'}
```

In [22]:
```
f.add('0')
print(f)
```

```
{0, 1, 2, 3, 4, 5, '5', 67, 7, '0', '56', '678'}
```

In [23]:
```
a={4,5,6,7,8}
b={4,5,11,22,778}
a.difference(b)
```

Out[23]:
```
{6, 7, 8}
```

In [24]:
```
b.difference(a)
```

Out[24]:
```
{11, 22, 778}
```

In [25]:
```
a={1,2,3}
b={5,6,7}
a.union(b)
```

Out[25]:
```
{1, 2, 3, 5, 6, 7}
```

In [27]:
```
b.union(a)
```

Out[27]:
```
{1, 2, 3, 5, 6, 7}
```

In [29]:
```
a={5,56,4,5,4,}
b={4,4,4,44}
a.union(b)
```

Out[29]:
```
{4, 5, 44, 56}
```

In [30]:
```
a.intersection(b)
```

Out[30]:
```
{4}
```

In [32]:
```
a.intersection(a)
```

Out[32]:
```
{4, 5, 56}
```

In [36]:
```
a={1,2,3,4}
b={}
b=a.copy()
print(b)
```

```
{1, 2, 3, 4}
```

In [38]:
```
a={1,2,3,4,5}
a.pop()
print(a)
```

```
{2, 3, 4, 5}
```

In [39]:
```
a.remove(5)
```

In [40]:
```
print(a)
```

```
{2, 3, 4}
```

In [41]:
```
a.remove(2,3)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-41-9f2229b4e39f> in <module>
----> 1 a.remove(2,3)

TypeError: remove() takes exactly one argument (2 given)
```

In [42]:
```
a.remove(2)
```

In [43]:
```
print(a)
```

```
{3, 4}
```

In [47]:
```
#isdisjoint()
a={'apple','microsoft','google','insta'}
b={'facebook','twitter','insta','whatsapp'}
z=a.isdisjoint(b)
print(z)
```

```
False
```

functions

In [5]:
```
a function is a group of related statements that performs a specific task.
functions are divided into two types.
1.Builtin functions:- min(),max(),len().....etc
2.User defined funcitons:- These functions are created by the users

```

In [ ]:
```
user defined functions are 4 types
1.with arguments with return value
2.with arguments without return value
3.without arguments and with return value
4.without arguments and without return value

```

In [ ]:
```
functioin syntax:
    def functionname(parameters/arguments):
        statements

```

In [ ]:
```

```