

```
In [4]: 1 # Adding a new contact
2 mycontacts={}
3 def addingContact(name,phoneNo):
4     if name not in mycontacts:
5         mycontacts[name]=phoneNo
6         print("new conatact %s is added" %name)
7     else:
8         print("%s is already in mycontacts"% name)
9     addingContact("siri","8878933739")
```

new conatact siri is added

```
In [7]: 1 # Searching for a Contact in the existing Contacts List
2 def searchMycontact(name):
3     if name in mycontacts:
4         print(name,":",mycontacts[name])
5     else:
6         print("%s doesn't exist"% name)
7     searchMycontact("siri")
```

siri : 8878933739

```
In [18]: 1 # Adding New Contacts to existing contacts list
2 # Merge another dictionary
3 def importNewcontacts(newContacts):
4     #if name not in mycontacts:
5     mycontacts.update(newContacts)
6     print(len(newContacts.keys()), " contacts are added")
7     # else:
8     #     print("Contacts are already in my contacts")
9
10 newContacts = { "pinni":8989898989,"babai":7878787878}
11
12 importNewcontacts(newContacts)
13 mycontacts
```

2 contacts are added

Out[18]: {'siri': '8878933739', 'pinni': 8989898989, 'babai': 7878787878}

```
In [ ]: 1 myContacts
```

```
In [25]: 1 # Edit a contact with new number
2 def EditContacts(name,phone):
3     if name in mycontacts:
4         mycontacts[name]=phone
5         print("%s contact has modified : " %name,mycontacts[name])
6         return
7
8     EditContacts("siri","1238292823")
```

siri contact has modified : 1238292823

```
In [26]: 1 # Remove a contact from existing contacts list
2 def RemoveAContact(name):
3     if name in mycontacts:
4         mycontacts.pop(name)
5         print(mycontacts)
6         return
7 RemoveAContact("siri")
```

```
{'pinni': 8989898989, 'babai': 7878787878}
```

Contacts Application

- Add,Search,List,,Modify Delete contacts

Find and Replace Application?

- Count the total number of occurrences of a word
- If word is existing
- Repalce all occurrences of a word with another word

Marks Analysis Application

- Generate marks file for n students (btw 1 to 100)
- Input : Marks text file - each line contains marks of one student
- Generates a report with the following information
 - Class Average
 - % of students passed
 - % of students failed
 - % of students with distinction
 - Highest Mark Frequency
 - Lowest Mark Frequency

```
In [22]: 1 # Function to generate marks data for n students
2 from random import randint
3
4 def generateMarks(n, lb, ub):
5     with open('DataFiles/marks.txt','w') as f:
6         for i in range(0, n):
7             r = randint(lb, ub)
8             f.write(str(r) + '\n')
9     return
10 generateMarks(100,0,100)
11
```

```
In [ ]: 1 s=input()
2 DigitFreq()
```

