`Dictionaries`

Creating a Dictionary

```
D1 = {}      #curly braces are used for dictionary
D1
```

```
{}
```

```
D2 = dict()
D2
```

```
{}
```

```
D3 = {'Ankush':95,'Arushi':89,'Rahul':92,'Abhay':85}
D3
```

```
{'Abhay': 85, 'Ankush': 95, 'Arushi': 89, 'Rahul': 92}
```

Accessing Elements in a Dictionary

```
D3
```

```
{'Abhay': 85, 'Ankush': 95, 'Arushi': 89, 'Rahul': 92}
```

```
D3["Arushi"]  #gives value corresponding to the key Ram
```

```
89
```

```
D3['Kritika']  #the key does not exist
```

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
<ipython-input-8-2bca8320b078> in <module>()
----> 1 D3['Kritika']  #the key does not exist

KeyError: 'Kritika'
```

SEARCH STACK OVERFLOW

Dictionaries are Mutable

```
D = {'Ankush':95,'Arushi':89,'Rahul':90,'Abhay':85}
D
```

```
{'Abhay': 85, 'Ankush': 95, 'Arushi': 89, 'Rahul': 90}
```

```
D['Kritika'] = 93
D
```

```
{'Abhay': 85, 'Ankush': 95, 'Arushi': 89, 'Kritika': 93, 'Rahul': 90}
```

```
D['Abhay'] = 96  #Marks of Abhay changed to 93.6
D
```

```
{'Abhay': 96, 'Ankush': 95, 'Arushi': 89, 'Kritika': 93, 'Rahul': 90}
```

Dictionay Operations : Membership

```
D
```

```
{'Abhay': 96, 'Ankush': 95, 'Arushi': 89, 'Kritika': 93, 'Rahul': 90}
```

```
'Rahul' in D
```

```
True
```

```
'Ankush' not in D
```

```
False
```

```
#NOTE: Membership test is only for keys not for values
```

Dictionary Operations: Transvering a Dictionary

```
#Dictionary tranversal using for loop:
D = {'Ankush':95,'Abhay':85,'Rahul':98,'Rinkal':85}

#Method 1:
for key in D:
    print(key,':',D[key])
```

```
Ankush : 95
Abhay : 85
Rahul : 98
Rinkal : 85
```

```
#Method 2:
for key,value in D.items():
    print(key,':',value)
```

```
Ankush : 95
Abhay : 85
Rahul : 98
Rinkal : 85
```

Method | Description | Example

```
#dict()     Creates a dictionary from a sequance of key-value pairs
L = [('Ankush',95),('Rahul',98),('Ujjwal',78),('Surbhi',89)]
L
```

```
[('Ankush', 95), ('Rahul', 98), ('Ujjwal', 78), ('Surbhi', 89)]
```

```
D = dict(L)
D
```

```
{'Ankush': 95, 'Rahul': 98, 'Surbhi': 89, 'Ujjwal': 78}
```

```
#keys()     Reutrns a list of keys in the dictionary
print(D)
D.keys()
```

```
{'Ankush': 95, 'Rahul': 98, 'Ujjwal': 78, 'Surbhi': 89}
dict_keys(['Ankush', 'Rahul', 'Ujjwal', 'Surbhi'])
```

```
#values()   Returns a list of values in the dictionay
print(D)
D.values()
```

```
{'Ankush': 95, 'Rahul': 98, 'Ujjwal': 78, 'Surbhi': 89}
dict_values([95, 98, 78, 89])
```

```
#items()    Returns a list of tuples(key-value)pair
print(D)
D.items()
```

```
{'Ankush': 95, 'Rahul': 98, 'Ujjwal': 78, 'Surbhi': 89}
dict_items([('Ankush', 95), ('Rahul', 98), ('Ujjwal', 78), ('Surbhi', 89)])
```

```
#get()      Returns the value corresponding to the key passed as the argument.
""" If the key is not present in the dictionry it will return None. """
print(D)
D.get('Ankush')
```

```
{'Ankush': 95, 'Rahul': 98, 'Ujjwal': 78, 'Surbhi': 89}
95
```

```
D.get('Ujjwal')
```

```
78
```

```
D.get('Rinkal')
```

```
#update()     appends the key value pair of the dictionay passed as the argument to
#             the key-value pair of the given dictionary.
D1 = {'Ankush':95,'Ujjwal':87,'Surbhi':98,'Rinkal':74}
D2 = {'Arushi':78,'Abhay':94}
D1.update(D2)
D1
```

```
{'Abhay': 94,
 'Ankush': 95,
 'Arushi': 78,
 'Rinkal': 74,
 'Surbhi': 98,
 'Ujjwal': 87}
```

```
D2
```

```
{'Abhay': 94, 'Arushi': 78}
```

```
#del()      Deletes the item with the given key.
#           To delete the dictionary from the memory we write:
#           del Dict name
D = {'Ankush':95,'Ujjwal':94,'Rahul':89,'Kritika':98}
del D['Ankush']
D
```

```
{'Kritika': 98, 'Rahul': 89, 'Ujjwal': 94}
```

```
del D
D
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-12-191bcbf31fad> in <module>()
      1 del D
----> 2 D

NameError: name 'D' is not defined
```

SEARCH STACK OVERFLOW

```
#clear()     Deletes or clear all the items of the dictionary
D = {'Ankush':95,'Ujjwal':98,'Kritika':89,'Surbhi':79}
D.clear()
D
```

Questions

```
''' Q. Create a dictionary'ODD' of odd numbers one and ten,where
```

```
key is the decimal number and the value is the corresponding numbers in words.
Perform the following operation on the dictionary:
a) Display the keys
b) Display the values
c) Display the items
d) Find the length of the dictionary
e) Check if 7 is present or not
f) Check if 2 is present or not
g) Retrieve the value corresponding to the key 9
h) Delete the items from the dictionary corresponding to the key 9 '''
```

```
#Solutin:
ODD = {1:'One',3:'Three',5:'Five',7:'Seven',9:'Nine'}
ODD
```

```
{1: 'One', 3: 'Three', 5: 'Five', 7: 'Seven', 9: 'Nine'}
```

```
# a) Display the keys
ODD.keys()
```

```
dict_keys([1, 3, 5, 7, 9])
```

```
# b) Display the values
ODD.values()
```

```
dict_values(['One', 'Three', 'Five', 'Seven', 'Nine'])
```

```
# c) Display the items
ODD.items()
```

```
dict_items([(1, 'One'), (3, 'Three'), (5, 'Five'), (7, 'Seven'), (9, 'Nine')])
```

```
# d)Find the length of the dictionary
len(ODD)
```

```
5
```

```
# e)Check if 7 is present or not
7 in ODD
```

```
True
```

```
# f)Check if 2 is present or not
2 in ODD
```

```
False
```

```
# g)Retrieve the values corresponding to the keys 9
ODD.get(9)
```

```
'Nine'
```

```
# h)Delete the item from the dictionary corresponding to the key 9
ODD = {1:'One',3:'Three',5:'Five',7:'Seven',9:'Nine'}
del ODD[9]
ODD
```

```
{1: 'One', 3: 'Three', 5: 'Five', 7: 'Seven'}
```

```
# Q.Write a program take names of employees and thier salaries as input and
# store them in a dictionary.

num = int(input("Enter the number of employees to be stored: "))
count = 1
employee = dict()  #create an empty dictionary
while count<= num:
    name = input("Enter the name of the Employee: ")
    salary = int(input("Enter the salary: "))
    employee[name] = salary
    count += 1
print("\n\nEMPLOYEE_NAME\tSALARY")
for i in employee:
    print(i,'\t\t',employee[i])
```

```
Enter the number of employees to be stored: 3
Enter the name of the Employee: Ankush
Enter the salary: 4000000
Enter the name of the Employee: Ujjwal
Enter the salary: 1400000
Enter the name of the Employee: Surbhi
Enter the salary: 3000000


EMPLOYEE_NAME   SALARY
Ankush           4000000
Ujjwal           1400000
Surbhi           3000000
```

```
# Write a program to count the nuber of times a character appears in a given string.

#count the number of times a character appears in a given string
st = input("Enter a string: ")
dic = {}  #creates an empty dictionary
for ch in st:
    if ch in dic: #if next character is already in D
        dic[ch] += 1
    else:
        dic[ch] = 1 #if ch appears for the first time
for key in dic:
    print(key,':',dic[key])
```

```
Enter a string: ankush rana
a : 3
n : 2
k : 1
u : 1
s : 1
h : 1
  : 1
r : 1
```

```python
# Write a function to convert a number entered by the user into its corresponding
# number name. e.g. if the input is 876 then the output should be 'Eight Seven Six'.


# Write a function to convert number into numbers names
def convert(num):
    #numbersNames is a dictionary of digits and correspondig number names
    numberNames = {0:'Zero',1:'One',2:'Two',3:'Three',4:'Four',5:'Five',6:'Six',7:'Seven',\
               8:'Eigth',9:'Nine'}
    result = ''
    for ch in num:
             key = int(ch) #converts character to integer
             value = numberNames[key]
             result = result+' '+value
    return result
num = input("Enter any number: ") #number is stored as string
result=convert(num)
print("The number is: ",num)
print("The numberName is: ",result)
```

```
Enter any number: 1912
The number is:  1912
The numberName is:   One Nine One Two
```

```python
month = {}
month[1] = 'Jan'
month[2] = 'Feb'
month[3] = 'Mar'
month[4] = 'Apr'
month
```

```
{1: 'Jan', 2: 'Feb', 3: 'Mar', 4: 'Apr'}
```

```python
type(month)
```

```
dict
```

```python
price = {'tomato':40,'cucumber':30,'potato':20,'cauliflower':70,'cabbage':50,'lettuce':40,'raddish':30,'carrot'
```

```python
price['potato']
```

```
20
```

```python
price['carrot']
```

```
20
```

```python
price.keys()
```

```
dict_keys(['tomato', 'cucumber', 'potato', 'cauliflower', 'cabbage', 'lettuce', 'raddish', 'carro
price.values()
```

```
dict_values([40, 30, 20, 70, 50, 40, 30, 20, 80])
```

```
price.items()
```

```
dict_items([('tomato', 40), ('cucumber', 30), ('potato', 20), ('cauliflower', 70), ('cabbage', 50
```

```
price['tomato'] = 25
price
```

```
{'cabbage': 50,
 'carrot': 20,
 'cauliflower': 70,
 'cucumber': 30,
 'lettuce': 40,
 'peas': 80,
 'potato': 20,
 'raddish': 30,
 'tomato': 25}
```

```
counting = {1:'one', 'one':1, 2:'two', 'two':2}
''' keys in a dictionary may be of heterogeneous types'''
```

```
digits = {0:'Zero',1:'One',2:'Two',3:'Three',4:'Four',5:'Five',6:'Six',7:'Seven',
          8:'Eigth',9:'Nine'}
```

```
len(digits)     #length operator len (number of key-value pairs in a dict)
```

```
10
```

```
digits[1]       #Indexing
```

```
'One'
```

```
min(digits)     #Function min
```

```
0
```

```
max(digits)     #Function max
```

```
9
```

```
sum(digits)     #Function sum(assuming keys are compatible for addition)
```

```
45
```

```
5 in digits     #Membership operator in
```

```
True
```

```
'Five' in digits
```

```
False
```

```
winter = {11:'November',12:'December',1:'January',2:'February'}
2 in winter, min(winter), max(winter), sum(winter)
```

```
(True, 1, 12, 26)
```

```
2 in winter.keys(), min(winter.keys()), max(winter.keys())\
,sum(winter.keys())
```

```
(True, 1, 12, 26)
```

```
del winter[11]
winter
```

```
{1: 'January', 2: 'February', 12: 'December'}
```

```
del winter
winter
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-45-075b7021c981> in <module>()
      1 del winter
----> 2 winter

NameError: name 'winter' is not defined
```

SEARCH STACK OVERFLOW

```
winter = {11:'November',12:'December',1:'January',2:'February'}
```

```
months = winter
```

```
months.clear()
months, winter
```

```
({}, {})
```

```
passwords = {'Ram':'ak@607','Shyam':'rou.589','Gita':'yam@694'}
passwords.get('Ram',-1)
```

```
'ak@607'
```

```
passwords.get('Raman',-1)
```

```
-1
```

```
print(passwords.get('Raman'))
```

```
None
```

```
morePasswords = {'Raman':'vi97@4','Kishore':'23@0jsk'}
passwords.update(morePasswords)
passwords
```

```
{'Gita': 'yam@694',
 'Kishore': '23@0jsk',
 'Ram': 'ak@607',
 'Raman': 'vi97@4',
 'Shyam': 'rou.589'}
```

```
#Function copy
morePasswords = {'Raman':'vi97@4','Kishore':'23@0jsk'}
newPasswords = morePasswords.copy()
id(newPasswords) , id(morePasswords)
```

```
(139850558843496, 139850558843640)
```

List of Function

```
D.items()
#Returns an object comprising of tuples of key-value pairs present in dict D
```

```
D.keys()
#Return an object comprising of all keys of dictionary D.
```

```
D.values()
#Returns an object comprising of all values of dict D.
```

```
D.clear()
#Removes all key-value pairs from  dict D.
```

```
D.get(key,default)
#For the specilied key, the function returns the associated value.
#Returns  the default value in the case key is not present in the dict D.
```

```
D.copy()
#Creates a shallow copy of dict D.
```

```
D1.update(D2)
#Adds the key-value pairs of dict D2 to dict D1.
```

Questions

```
''' Q. Use dictionary to store the marks of the students in four sunjects.
Write a function to find the name of the student scoring highest percentage.'''


def highest():
    max_p = 0
    name = " "
    for i,j  in student.items():
        if max_p<sum(j):
            max_p=sum(j)
            name=i
    result=max_p/4
    print("highest percentage is scored by",name," ",result)
student={'Ankush':[95,85,78,98],'Ujjwal':[87,67,98,78],'Rinkal':[89,67,87,90],
         'Rahul':[79,87,77,97]}
print("name of the students and marks scored by them in each subject:")
for key in student:
    print(key,':',student[key])
highest()
```

```
name of the students and marks scored by them in each subject:
Ankush : [95, 85, 78, 98]
Ujjwal : [87, 67, 98, 78]
Rinkal : [89, 67, 87, 90]
Rahul : [79, 87, 77, 97]
highest percentage is scored by Ankush   89.0
```

```
''' Q. Create two dictionaries odd and even where the keys is decimal number and
the value is corresponding number in word.

Perform the following operation:-
1) Find total number of odd numbers.
2) Accpet 10 numbers from the user if number is odd put
it in odd dictionary otherwise put in it even dictionary.
3) Find sum of all even numbers.'''

odd = {}
even = {}
c = 0
check_even = {0:'Zero',2:'Two',4:'Four',6:'Six',8:'Eight'}
check_odd = {1:'One',3:'Three',5:'Five',7:'Seven',9:'Nine'}
print("Enter 10 numbers: ")
for i in range(1,11):
  k = int(input("Enter number "+str(i)+": "))
  if k%2==0:
    for key in check_even:
      if k==key:
        even.update({k:check_even[key]})
        break
      else:
        if k%2!=0:
          for key in check_odd:
            if k==key:
              odd.update({k:check_odd[key]})
              c += 1
              break
              print(odd)
print("Total numbers of odd nubers: ",c)
sum = 0
for key in even:
  sum += key
print(sum)
```

```
Enter 10 numbers:
Enter number 1: 1
Enter number 2: 2
Enter number 3: 3
Enter number 4: 0
Enter number 5: 4
Enter number 6: 5
Enter number 7: 6
Enter number 8: 7
Enter number 9: 8
Enter number 10: 9
Total numbers of odd nubers:  0
20
```

```
'''Q. Write a function that takes a sentence as input from the user and
calculates the frequency of each letter.
Use a Variable of Dictionary type to maintain the count.'''

user_str = input("Enter a string:")
dict = {}
for char in user_str:
    if char in dict:
        dict[char] += 1
    else:
        dict[char] = 1
for key in dict:
    print(key,':',dict[key])
```

```
Enter a string:ankush rana
a : 3
n : 2
k : 1
u : 1
s : 1
h : 1
  : 1
r : 1
```

```
'''Q. Create a dictionary sunbj_stud that maps a list of students to
the subject they are studying as per the following information:
Write statement for finding the subject with the minimum
number of students and removing those subjects from subj_stud.'''

sub = {"Maths":["Joe","Sue","Ben"],"Physics":["Joe","Mike","Michael"],"Biology":["Sue","John"],"Computers":["Jo
min = None
l = []
for i,j in sub.items():
    if min==None or min>len(j):
     l = [i]
     min = len(j)
    elif min==len(j):
      l.append(i)
for i in l:
    del sub[i]
print(sub)
```

```
{'Maths': ['Joe', 'Sue', 'Ben'], 'Physics': ['Joe', 'Mike', 'Michael']}
```

```
subj_stud = {'Maths':['Joe','Sue','Ben'],'Physics':['Joe','Mike','Micheal']}
subj_stud['Biology'] = ['Sue','John']     #Dictionary is mutable
subj_stud['Computers'] = ['John','Chris']  #Dictionary is mutable
print(subj_stud)
min = None
l = []
for i,j in subj_stud.items():
    if min == None or min>len(j):
        l = [i]
        min = len(j)
    elif min == len(j):
        l.append(i)
for i in l:
    del subj_stud[i]
print(subj_stud)
```

```
{'Maths': ['Joe', 'Sue', 'Ben'], 'Physics': ['Joe', 'Mike', 'Micheal'], 'Biology': ['Sue', 'John'
{'Maths': ['Joe', 'Sue', 'Ben'], 'Physics': ['Joe', 'Mike', 'Micheal']}
```

```
'''Q. write a python function that prints a dictionary where the keys are numbers
b/w 1 & 5 and the values are cubes of the keys.'''

num = int(input("Enter the no. of cubes to be stored: "))
count = 1
cubes = dict()
while count <= num:
        numbers = int(input("Enter the no. whose cube is to be find: "))
        cube = numbers**3
        cubes[numbers] = cube
        count += 1
print(cubes)
```

```
Enter the no. of cubes to be stored: 5
Enter the no. whose cube is to be find: 1
Enter the no. whose cube is to be find: 2
Enter the no. whose cube is to be find: 3
Enter the no. whose cube is to be find: 4
Enter the no. whose cube is to be find: 5
{1: 1, 2: 8, 3: 27, 4: 64, 5: 125}
```