In [ ]:
```
#DICTIONARY QUESTIONS
```

In [10]:
```
"""Write a program that repeatedly asks the
user to enter
product names and prices.
Store all of these
in a dictionary
whose keys are the product names and whose
values are the prices.
When the user is done entering products and prices,
allow them to repeatedly enter a product
name and print the corresponding price or a message
if the product is not in the dictionary."""
#Key are the products
#values are the prices
price={}#the list will store the dictionary keys
def store():
    product={}#empty dictionary
    #price={}#the list will store the dictionary keys
    #create an empty dictionary
    while True:
        product=input("enter product:")
        if product=='q':
            break
        price[product]=int(input("enter the price:"))
        if price == 'q':
            break
    print(price)
    #alllow a user to enter the name of item and display price
    i=2
    while i<2:
        pro=input("enter product to check the store:")
        for p in price:
            if pro<=p:
                print(p,price[p])
            else:
                print("product not in the store")
def dollar():
    dolla=int(input("Enter some dollar to display items"))
    print("This are products below{} dollars\n".format(dolla))
    for p in price:
        if dolla==price[p]:
            print(p,price[p])

store()
dollar()
```

```
enter product:egg
enter the price:100
enter product:kiko
enter the price:3
enter product:milk
enter the price:60
```

```
enter product:rice
enter the price:500
enter product:sugar
enter the price:400
enter product:milo
enter the price:3
enter product:fat
enter the price:20
enter product:q
{'egg': 100, 'kiko': 3, 'milk': 60, 'rice': 500, 'sugar': 400, 'milo': 3, 'fat': 20}
Enter some dollar to display items100
This are products below100 dollars

egg 100
```

In [2...

```
"""
 Write a program that uses a dictionary that
 contains ten user names and passwords.
 The program should ask the user to enter their username and password.
 If the username is not in the dictionary,
 the program should indicate that thepersonisnotavaliduserofthesystem.
 If theusernameisinthedictionary,buttheuserdoesnotentertherightpassword,
 theprogram should say that the password is invalid. If the password is correct,
 then the program should tell the user
 that they are now logged in to the system.
"""
#the program contains 10 user names and passwords
#the program should ask the user to enter password and user name
#if the user name is not in dictionary indicate inalid
#if the user name is in the dictionary but wrong password indicate invalid
def login_system():
    account={'flavian':'123','anselmo':'000','leon':'111'}
    #the dictionary with user names as the key and passwords as values
    user_input=input("Enter user_name:")
    pwd_input=input("Enter password:")
    for key,value in account.items():
        if key==user_input and value==pwd_input:
            print("log_in successfull!")
            break

        else:
            print("Wrong password or user_name!")

login_system()
#error encountered
"""
valueerror: too many values to unpack (expected 2)
A value error is raised when you try to access information
from a value that does not exist
values can be any object such as alist
```

```
      In python "unpacking" refers to retrieving items from a value
      for example retrieving items from a file is called unacking
      ->the error above tells us that we are trying to unpack
      too many values from a value
      ->This error occurswhen we iterate oer a dictionary and unpack
      its keys and values seperatly
      or
      ->When you forget to unpack every item from a list to a variable.
      for key, value in hydrogen:
              print("Key:", key)
              print("Value:", str(value))
          ->in this code we unpack "hydrogen" into values key and values
          ->we want "key" to correspond to the keys in our dict
          ->and "values" to correspond to the values
      #to solve this problem use method item()
      #item()->This method analyzes a
      dictionary and returns keys
      and values stored as tuples.
      Let’s add this method to our code:
      We have added the items() method to the end of “hydrogen”.
      This returns our dictionary with key-value pairs stored as tuples.
      We can see this by printing out
      the contents of hydrogen.items() to the console:

      for key, value in hydrogen.items():
              print("Key:", key)
              print("Value:", str(value))
      """

Enter user_name:leon
Enter password:111
Wrong password or user_name!
Wrong password or user_name!
log_in successfull!


('\nvalueerror: too many values to unpack (expected 2)\nA value error is raised when yo
 u try to access information\nfrom a value that does not exist \nvalues can be any obje
 ct such as alist\n\nIn python "unpacking" refers to retrieving items from a value \nfo
 r example retrieving items from a file is called unacking \n->the error above tells us
 that we are trying to unpack \ntoo many values from a value \n->This error occurswhen
 we iterate oer a dictionary and unpack \nits keys and values seperatly \nor\n->When yo
 u forget to unpack every item from a list to a variable.\nfor key, value in hydroge
 n:\n\tprint("Key:", key)\n\tprint("Value:", str(value))\n    ->in this code we unpack
 "hydrogen" into values key and values\n    ->we want "key" to correspond to the keys i
 n our dict\n    ->and "values" to correspond to the values \n#to solve this problem us
 e method item()\n#item()->This method analyzes a \ndictionary and returns keys \nand v
 alues stored as tuples. \nLet’s add this method to our code:\nWe have added the item
 s() method to the end of “hydrogen”.\nThis returns our dictionary with key-value pairs
 stored as tuples. \nWe can see this by printing out \nthe contents of hydrogen.items()
 to the console:\n\nfor key, value in hydrogen.items():\n\tprint("Key:", key)\n\tprin
 t("Value:", str(value))\n')
```

In [28]:
```
account={'flavian':'123','anselmo':'000','leon':'111'}
for i,j in account.items():
    print(i+j)
```

```
flavian123
anselmo000
leon111
```

In [ ]:

In [ ]:

In [ ]: