Computational Thinking and Programming - 2

Dictionaries

XI

Definition

Python dictionary is an unordered collection of items. While other compound data types have only value as an element, a dictionary has a **key: value** pair. An item in the dictionary has a key and the corresponding value expressed as a pair, key : value.

- No two words (keys) in the dictionary are the same, but this restriction does not apply to their values (meanings).
- While values can be of any data type and can repeat, keys must be of immutable type (string, number or tuple with immutable elements) and must be unique.
- Dictionary keys are case sensitive, same name but different cases of keys will be treated distinctly.

Examples of Dictionaries

Example :

key value

Student = {“rollno”:101,”name”:”Rahul”,”marks”:89}

scores = { 'Sunita' : 90 , 'Rahul' : 78 , 'Meena' : 80 }

capitals = {"USA":"Washington D.C.", "France":"Paris", "India":"New Delhi"}

students={1: **{'name': 'Steve', 'age': 25, 'marks': 60}**, 2: {'name': 'Anil', 'age': 23, 'marks': 75}}

key value

Creating dictionaries

- **Creating an empty dictionary**

```
d = {}
    OR
d= dict()
```

- **Creating and initialising a dictionary**

```
my_dictionary = {1: 'apple', 2: 'ball'}
my_dictionary = {"January":31, "February":28, "March":31, "April":30,"May":31}
d = {'Name': 'Hamza', “Events”: [“quiz”,”hindi quiz”,”warp”,”english debate”]}
```

Creating dictionaries

Note : **Remember the keys must be of immutable data type only.**

The following statement shows an error :

Here key is a list which is mutable

```
>>> d = {[1,2]:30,"one":40}
    Traceback (most recent call last):
    File "<pyshell#9>", line 1, in <module>
    d = {[1,2]:30,"one":40}
    TypeError: unhashable type: 'list'
```

Creating dictionaries

Creating a dictionary from name and value pairs:

- Specifying **Key: Value** pairs as arguments to dict() function :

  Keys are passed as arguments and values as the values of the arguments.

  ```
  >>> employee=dict (name = 'John', salary=10000, age=24)
  ```

- Specifying comma separated **key:value** pairs

  Key and value pairs are enclosed in curly brackets

  ```
  >>>employee=dict ({"name" : 'John', "salary":10000, "age":24})
  ```

Creating dictionaries

➢ Specify keys separately and corresponding values separately

Keys and values are enclosed separately as tuples and are given as arguments to zip() function

```
>>>employee=dict(zip(("name","salary","age"),("john",10000,24)))
```

➢ Specifying Key: Value pairs separately in form of sequences

A list or tuple is passed as an argument to dict() where each element of the sequence is a key:value pair

```
>>>employee=dict ([["name",'John'], ["salary", 10000], ["age",24] ] )
>>>employee=dict (([["name",'John'], ["salary", 10000], ["age",24] ) )
```

Accessing elements of dictionary:

The keys are used as indices to access the elements in a dictionary.

For example, the following statement

```
scores = { 'Sunita' : 90 , 'Rahul' : 78 , 'Meena' : 80 }
```

assigns the dictionary of three elements to the variable scores.

The keys, Sunita, Rahul and Meena are used as indices to access the values, 90, 78 and 80 respectively.

Accessing elements of dictionary:

- Elements can be accessed using a for loop where the range is specified by keys() method of the dictionary.

```
d= {"January":31, "February":28, "March":31, "April":30}
for k in d.keys():
  print(k, ' has ',d[k], ' days ')
```

# each iteration of the for loop refers to a key

Accessing elements of dictionary:

- ❖ The method items() in the dictionary iterates over all the keys and helps us to access the key-value pair one after the other in the loop and is also a good method to access dictionary keys with value.

```
d= {"January":31, "February":28, "March":31, "April":30}
for k,v in d.items():
      print(k, ' has ', v, ' days.')
```

# each iteration of the for loop retrieves a tuple

- ❖ Display the whole dictionary at once using the items() method

```
print(d.items())
```

Operations on DICTIONARIES

OPERATIONS

1. Adding Items To Dictionary
2. Changing Items In Dictionary
3. Deleting Items From Dictionary

Adding Items to Dictionary

A new item (key:value pair) can also be added by the simple assignment statement. The only constraint is that the key must be unique. If the key already exists, then the value is simply changed as in the assignment statement above.

Syntax : dictionary name[new keyname]=new value

Example

```
fruits={1:"Apple",2:"Mango"}
print("Dictionary1 :")
print(fruits)
print("After adding :")
fruits[3]="Orange"
print(fruits)
```

OUTPUT

```
Dictionary1 :
{1:"Apple",2:"Mango"}
After adding :
{1:"Apple",2:"Mango",3:"Orange"}
```

Changing Items to Dictionary

An existing item (key:value pair) can also be changed or modified by the simple assignment statement.

Syntax : dictionary name[keyname]=new value

Example

```
fruits={1:"Apple",2:"Mango"}
print("Dictionary1 :")
print(fruits)
print("After changing :")
fruits[1]="Orange"
print(fruits)
```

OUTPUT

```
Dictionary1 :
{1:"Apple",2:"Mango"}
After adding :
{1:"Orange",2:"Mango"}
```

Creating a dictionary with input from the user

```
c = { }
for i in range(2):
    cname =input("Enter  country name : ")
    cap=input("Enter capital :")
    c[cname]=cap
print(c)
```

OUTPUT

```
Enter  country name : India
Enter capital :New Delhi
Enter  country name :Pakistan
Enter capital :Islamabad
{'India': 'New Delhi',
'Pakistan': 'Islamabad'}
```

```
c = { }
for i in range(2):
    cname =input("Enter  country name : ")
    cap=input("Enter capital :")
    c[cname]=cap
for i in c:
    print(c[i]," is capital of" ,i)
```

Exercise :

Consider the following dictionary and give the output:

weight={'A1':8,'A2':7,'B1':6,'B2':5,'C1':4,'C2':3,'D1':2,'D2':1,'E':0}

```
(i)  for k in weight:
          print(k,"-",weight[k])

(ii)  for k in weight.keys():
          print(k,"-",weight[k])

(iii)  for k in weight.values():
          print(k, end=' ')

(iv)  for k,v in weight.items():
          print(k,"-",v)
```

SOLUTION

(i)
A1 - 8
A2 - 7
B1 - 6
B2 - 5
C1 - 4
C2 - 3
D1 - 2
D2 - 1
E - 0

(ii)
A1 - 8
A2 - 7
B1 - 6
B2 - 5
C1 - 4
C2 - 3
D1 - 2
D2 - 1
E - 0

(iii) 8 7 6 5 4 3 2 1 0

(iv)
A1 - 8
A2 - 7
B1 - 6
B2 - 5
C1 - 4
C2 - 3
D1 - 2
D2 - 1
E - 0

Exercise

1. Create a dictionary with keys as number and values as their squares till n
2. Create a dictionary of n elements , with keys as number and values as a list/tuple of all factors of that number.
3. Create a dictionary of n elements , with keys as number and values as a list/tuple of five multiples of that number.
4. Create a dictionary of n elements, and values as "Prime" or "Composite" depending on key.

Solution

1.

```
n=int(input(“Enter the value of n :”))
d={}
for i in range(1,n+1);
    d[i]=i*i
print(d)
```

2.

```
n=int(input("Enter the value of n :"))
d={}
for i in range(1,n+1):
    t=()
    for j in range(1,i+1):
        if i%j==0:
            t=t+(j,)
    d[i]=t
print(d)
```

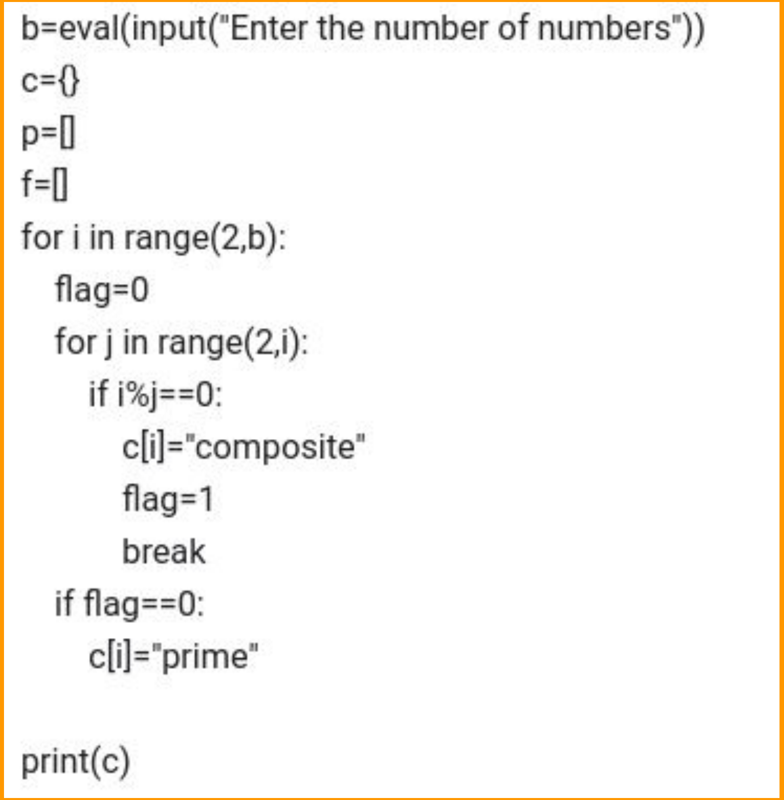
Solution

3.

```
n=int(input("Enter the value of n :"))
d={}
for i in range(1,n+1):
    t=()
    for j in range(1,6):

            t=t+(i*j,)
    d[i]=t
print(d)
```

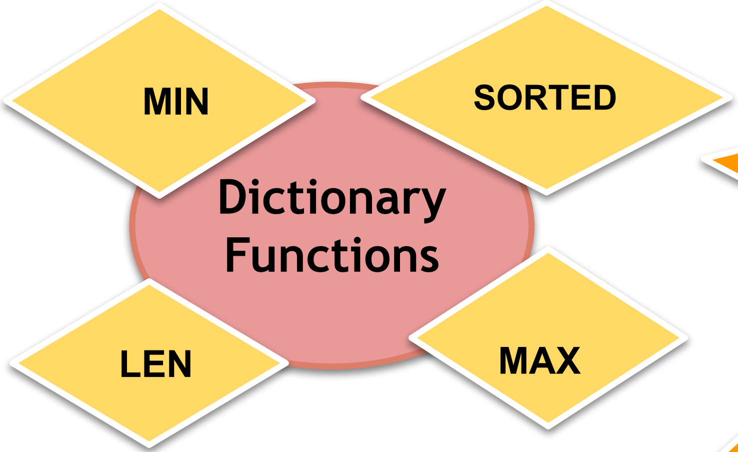
4.

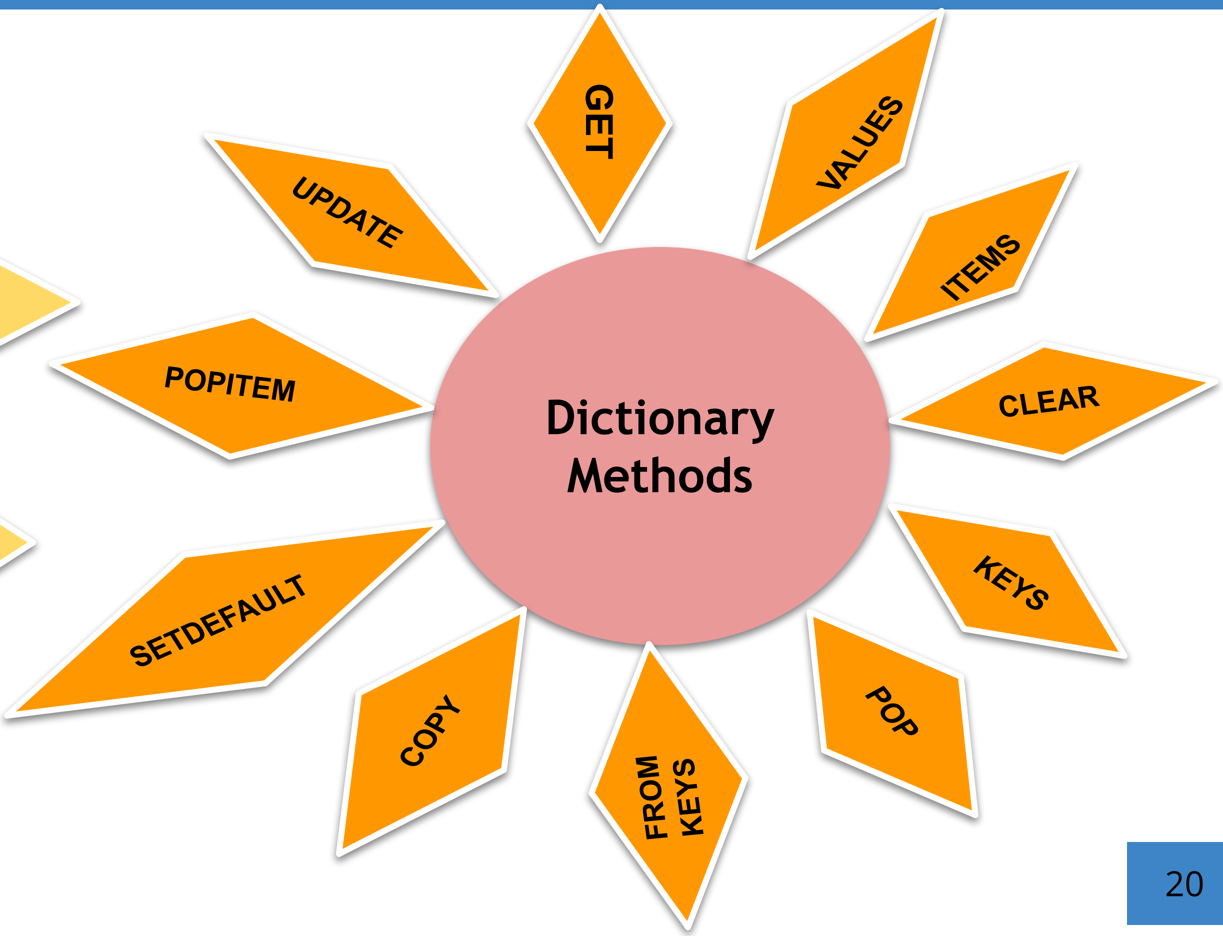


```
b=eval(input("Enter the number of numbers"))
c={}
p=[]
f=[]
for i in range(2,b):
    flag=0
    for j in range(2,i):
        if i%j==0:
            c[i]="composite"
            flag=1
            break
    if flag==0:
        c[i]="prime"

print(c)
```

Dictionary Functions and Methods





Dictionary Functions and Methods

1. **len()**

Returns the length of the dictionary or the number of elements in the dictionary

Syntax : len(dictionaryname)

Example

```
my_dict = {"January":31, "February":28}
print("Length:",len(dict1))
```

OUTPUT

```
Length : 2
```

Tuple Functions and Methods

2. max()

Returns the largest key from the dictionary. Keys must be comparable.

Syntax : max(dictionaryname)

Example

```
 D1={1:"Apple",2:"Mango"}
print("Maximum:",max(D1))
```

OUTPUT

```
Maximum: 2
```

Tuple Functions and Methods

3. min()

Returns the smallest key from the dictionary. Keys must be comparable.

Syntax : min(dictionaryname)

Example

```
D1={1:"Apple",2:"Mango"}
print("Minimum:",min(D1))
```

OUTPUT

```
Minimum: 1
```

Dictionary - sorted()

4. sorted()

It returns the list in sorted order (ascending by default). The dictionary passed as parameter remains unaffected.

Syntax : sorted(<dictionary>,<reverse = True/False (default)>)

Example

```
d1={1:"Jan",7:"Aug",4:"Apr"}
print("Sorted values:")
print(sorted(d1))
```

OUTPUT

```
Sorted values:
[1,4,7]
```

Dictionary - fromkeys()

6. fromkeys()

It creates a dictionary from a collection of keys (tuple/list/string) with a default value for all (if given) or a value None assigned to all of them.

Syntax : dict.fromkeys(seq)

```
e=["empid","name","salary"]
d1=dict.fromkeys(e)
print(d1)
d2=dict.fromkeys(e,"*")
print(d2)
```

OUTPUT

```
{'empid': None, 'name': None, 'salary': None}
{'empid': '*', 'name': '*', 'salary': '*'}
```

Dictionary - get()

7. **get()**

It returns the value for the given key. If key is not present , it returns None or Default value (if specified).

Syntax : dictionaryname.get(keyname)

Example

```
d1={1:"India",2:"Pak",7:"Srilanka"}
print("Value :")
print(d1.get(7))
print(d1.get(10))
print(d1.get(12,"Not available"))
```

OUTPUT

```
Value :
Srilanka
None
Not available
```

Dictionary - keys()

8. keys()

It returns a list of the keys of the given dictionary

Syntax : dictionaryname.keys()

Example

```
d1={1:"India",2:"Pak",7:"Srilanka"}
print("Keys of dictionary:")
print(d1.keys())
```

OUTPUT

```
Keys of dictionary:
dict_keys([1,2,7])
```

Dictionary - values()

8. values()

It returns a list of all the values of the given dictionary.

Syntax : dictionaryname.values()

```
d1={1:"India",2:"Pak",7:"Srilanka"}
print("Values:")
print(d1.values())
```

OUTPUT

```
Values :
dict_values(["India","Pak
","Srilanka"])
```

Dictionary - clear()

9. clear()

It removes all the items(key and value pairs) of the given dictionary

Syntax : dictionaryname.clear()

Example

```
d1={5:"May",2:"Feb",7:"Aug"}
d1.clear()
print("After clearing")
print(d1)
```

OUTPUT

```
After clearing :
{}
```

Dictionary - pop()

10. pop()

It removes item with provided key of the given dictionary and returns the value.

Syntax : dictionaryname.pop(keyname)

Example

```
s1={5:"Books",2:"Pencils",7:"Eras
er"}

print(d1.pop(2))
print("After popping")
print(d1)
```

OUTPUT

```
"Pencils"
After popping :
{5:"Books",7:"Eraser"}
```

Dictionary - popitem()

11. popitem()

It removes the last item from the given dictionary and returns the key:value pair as tuple. It returns a KeyError in case the dictionary is empty.

Syntax : dictionaryname.popitem()

Example

```
s1={5:"Books",2:"Pencils",7:"Eraser"}
print(s1.popitem())
print("After popping ")
print(s1)
s1.clear()
s1.popitem()
```

OUTPUT

```
(7:"Eraser")
After popping :
{5:"Books",2:"Pencils"}
KeyError
```

Dictionary - copy()

12. copy()

This method returns a copy of the dictionary. It doesn't modify the original dictionary.

Syntax : dictionaryname.copy()

Example

```
d1={1:"Jan",2:"Feb",3:"Mar"}
d2=d1.copy()
print(d1)
print("After using copy()")
print(d2)
```

OUTPUT

```
{1:"Jan",2:"Feb",3:"Mar"}
After using copy()
{1:"Jan",2:"Feb",3:"Mar"}
```

Dictionary - setdefault()

13. setdefault()

It returns the value of the key if it is in dictionary, and None if it is not dictionary.If key does not exist in the dictionary, default value is specified then it becomes key's value.

Syntax : dictionaryname.setdefault(keyname,default value)

Example

```
p= {'name': 'Phill', 'salary': 15000}
nm=p.setdefault('name','name not available')
gr=p.setdefault('gender','Not Available')
print("Employee Name : ", nm)
print("Gender : ",gr)
print(p)
```

OUTPUT

```
Employee Name :  Phill
Gender :  Not Available
{'name': 'Phill', 'salary': 15000,
'gender': 'Not Available'}
```

Dictionary - update()

14. update()

This method adds element(s) to the dictionary if the key is not in the dictionary. If the key is in the dictionary, it updates the key with the new value.The update() method takes either a dictionary or an iterable object of key/value pairs (generally tuples).

Syntax : dictionaryname.update(dictname/key:value pair)

Example

```
d1={1:"India",2:"USA",3:"JAPAN"}
print("After updating")
d1.update({2:"Canada"})
print(d1)
d1.update({4:"UK"})
print(d1)
```

OUTPUT

```
print("After Updating")
{1:"India",2:"Canada",3:"JAPAN"}
{1:"India",2:"Canada",3:"JAPAN",4:"UK"}
```

Dictionary - del statement

del statement

This statement deletes the specified key:value pair or entire dictionary.

Syntax : **del dictionaryname**
del dictionaryname[keyvalue]

Example

```
d1={1:"India",2:"USA",3:"JAPAN"}
del d1[2]
print("After deleting")
print(d1)
del d1
print(d1)
```

OUTPUT

```
After deleting"
d1={1:"India",3:"JAPAN"}
NameError
```

Exercise :

Find the output :

```
D=  {'U1':'Programming', 'U2':'Data structures','U3':'Files',
     'U4':'RDBMS', 'U5':'Networks'}

a)   print(len(D))
b)   print(D.pop('U1'))
c)   print(D.popitem())
d)   print(D.pop('U3'))
e)   print(D.get('U4'))
f)   print('U2' in D)
g)   print('Programming' in D)
h)   print('U3' not in D)
i)   print('Files' not in D)
```

Solution

a) 5
b) Programming
c) ('U5', 'Networks')
d) Files
e) RDBMS
f) True
g) False
h) False
i) True

Program #1

Write a program to accept roll number, names and marks of five students and store the detail in dictionary using roll number as key. Also display the sum of marks of all the five students.

s={1:("arnav",77,88,99,67,98).....}

Solution

```
students={}
for i in range(5):
    rollno=int(input("Enter the rollno:"))
    sdetails=eval(input("Enter name and marks of five subjects:"))
    students[rollno]=sdetails
print(students)
for i in students:
    sum=0
    for j in range(1,6):
        sum=sum+students[i][j]
    print("Total marks of student having rollno ",i,":",sum)
```

Program #2

To create a dictionary employee with the keys:

a. name: Name of the employee
b. basic: basic salary of the employee (an integer)
c. dept: Department (A string)

Then add the following keys to the dictionary:

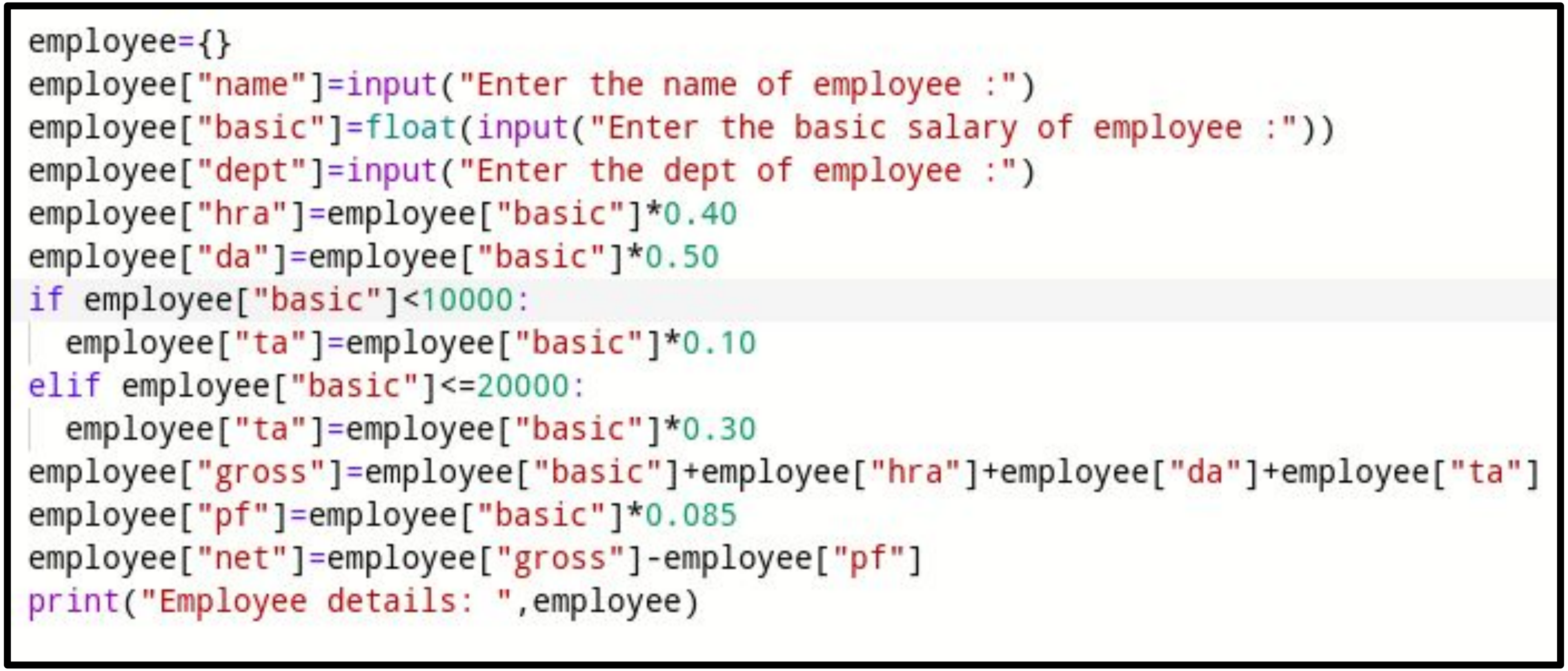
d. hra: House Rent Allowance (float – 40% of the basic)
e. da: Dearness Allowance (float – 50% of the basic)
f. ta : float – to be calculated as follows:

basic	ta
If less than 10000	10% of the basic
If >= 10000 and < = 20000	30% of the basic

g. gross: Gross salary (float – basic+hra+da+ta)
h. pf: Provident Fund (float – 8.5% of the basic)
i. net: Net salary (float – Gross salary-Provident Fund)

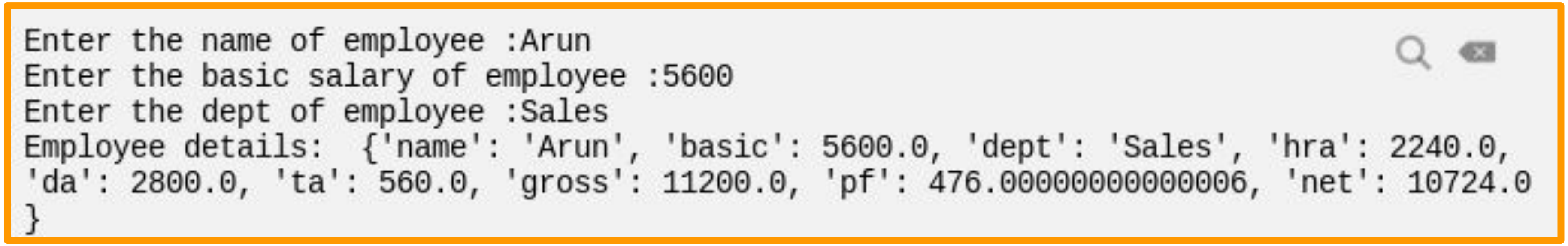
Input the values of name, basic, and dept from the user and calculate the values of other keys. Then display all the elements of employee in different lines.

Solution



```python
employee={}
employee["name"]=input("Enter the name of employee :")
employee["basic"]=float(input("Enter the basic salary of employee :"))
employee["dept"]=input("Enter the dept of employee :")
employee["hra"]=employee["basic"]*0.40
employee["da"]=employee["basic"]*0.50
if employee["basic"]<10000:
  employee["ta"]=employee["basic"]*0.10
elif employee["basic"]<=20000:
  employee["ta"]=employee["basic"]*0.30
employee["gross"]=employee["basic"]+employee["hra"]+employee["da"]+employee["ta"]
employee["pf"]=employee["basic"]*0.085
employee["net"]=employee["gross"]-employee["pf"]
print("Employee details: ",employee)
```

OUTPUT



```
Enter the name of employee :Arun
Enter the basic salary of employee :5600
Enter the dept of employee :Sales
Employee details:  {'name': 'Arun', 'basic': 5600.0, 'dept': 'Sales', 'hra': 2240.0,
'da': 2800.0, 'ta': 560.0, 'gross': 11200.0, 'pf': 476.00000000000006, 'net': 10724.0
}
```

Program #3

To create a dictionary employee with the keys:
a. name: Name of the employee
b. basic: basic salary of the employee (an integer)
c. dept: Department (A string)

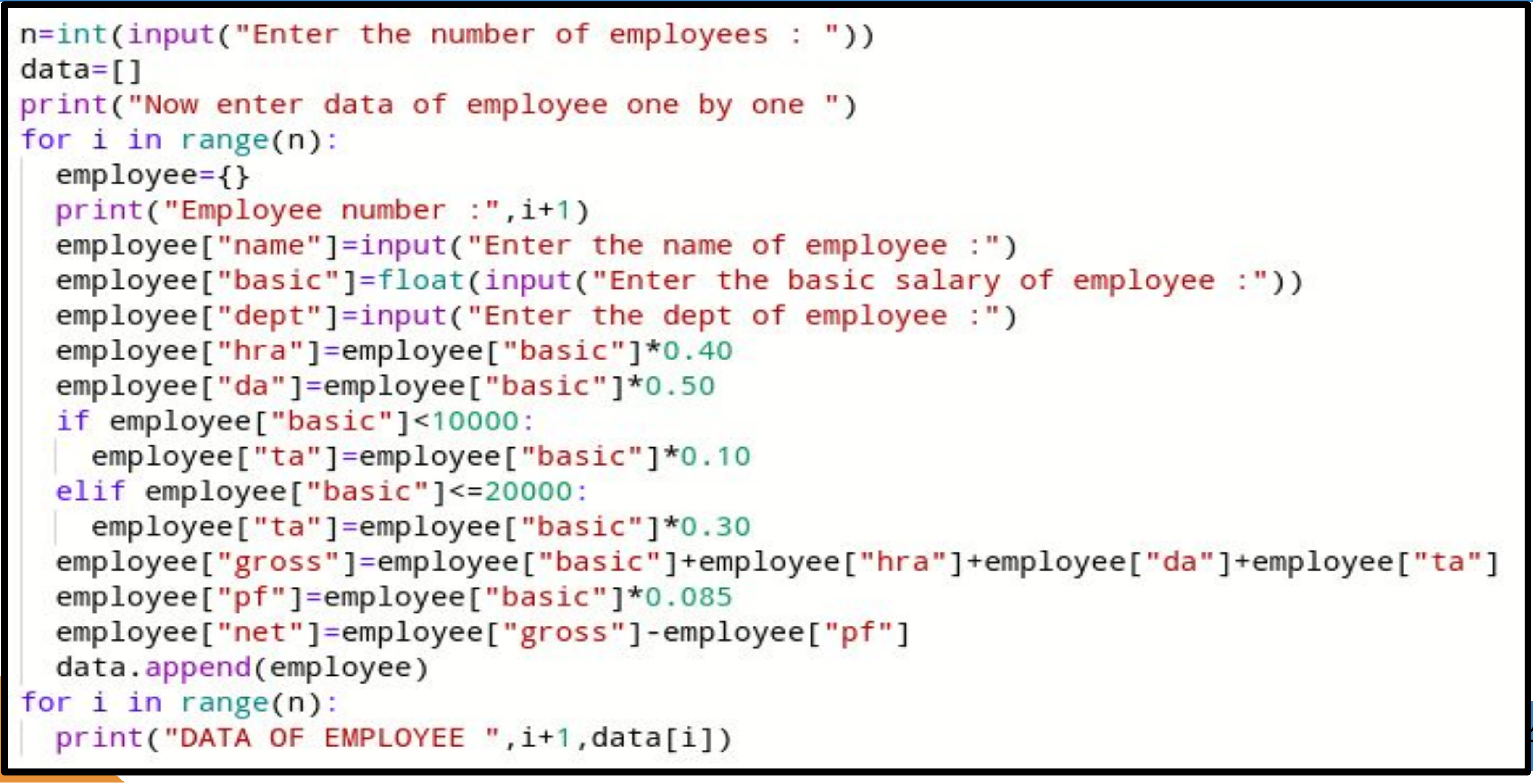
Then add the following keys to the dictionary:
d. hra: House Rent Allowance (float – 40% of the basic)
e. da: Dearness Allowance (float – 50% of the basic)
f. ta: float – to be calculated as follows:

basic	ta
If less than 10000	10% of the basic
If >= 10000 and < = 20000	30% of the basic

g. gross: Gross salary (float – basic+hra+da+ta)
h. pf: Provident Fund (float – 8.5% of the basic)
i. net: Net salary (float – Gross salary-Provident Fund)

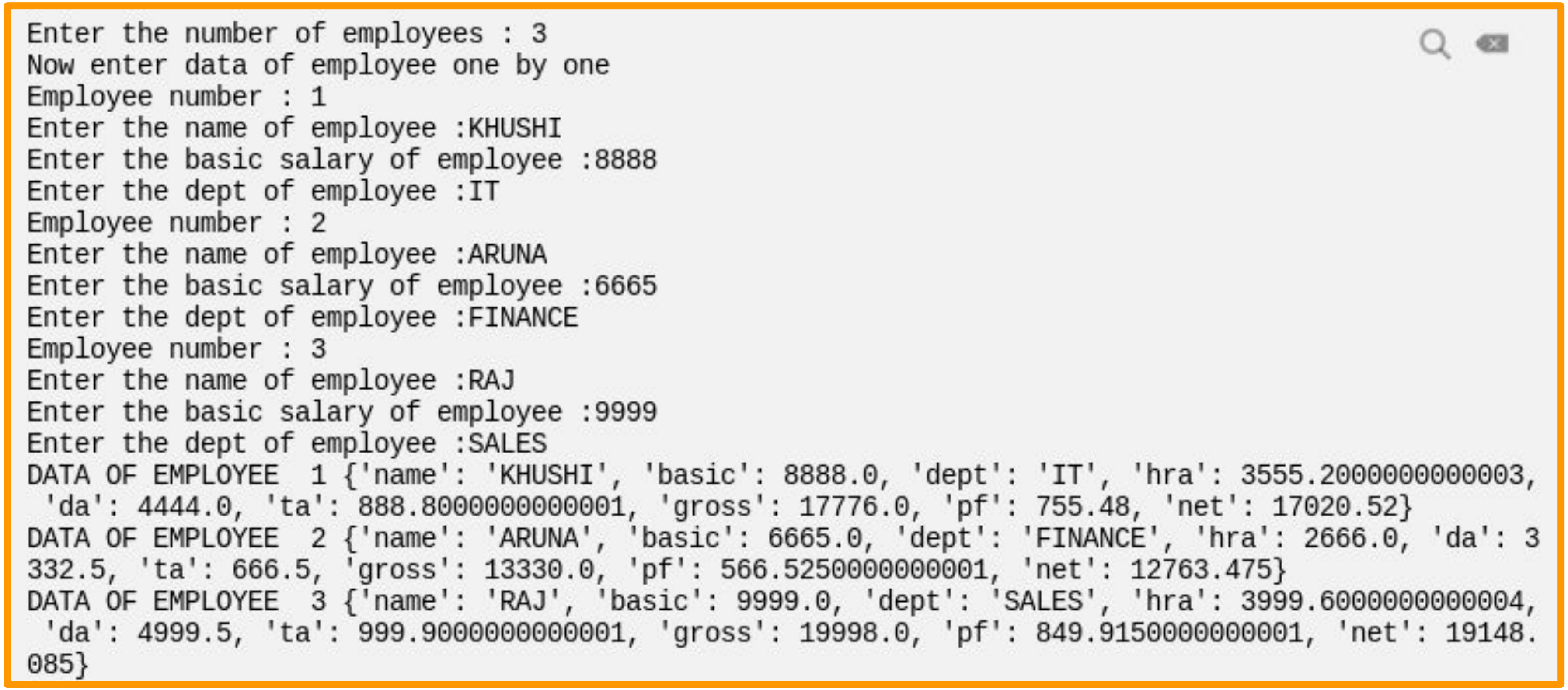
To create a list of employee dictionaries (as created above). The list can have n elements. Then display the details of all employees.

Solution



```python
n=int(input("Enter the number of employees : "))
data=[]
print("Now enter data of employee one by one ")
for i in range(n):
    employee={}
    print("Employee number :",i+1)
    employee["name"]=input("Enter the name of employee :")
    employee["basic"]=float(input("Enter the basic salary of employee :"))
    employee["dept"]=input("Enter the dept of employee :")
    employee["hra"]=employee["basic"]*0.40
    employee["da"]=employee["basic"]*0.50
    if employee["basic"]<10000:
        employee["ta"]=employee["basic"]*0.10
    elif employee["basic"]<=20000:
        employee["ta"]=employee["basic"]*0.30
    employee["gross"]=employee["basic"]+employee["hra"]+employee["da"]+employee["ta"]
    employee["pf"]=employee["basic"]*0.085
    employee["net"]=employee["gross"]-employee["pf"]
    data.append(employee)
for i in range(n):
    print("DATA OF EMPLOYEE ",i+1,data[i])
```

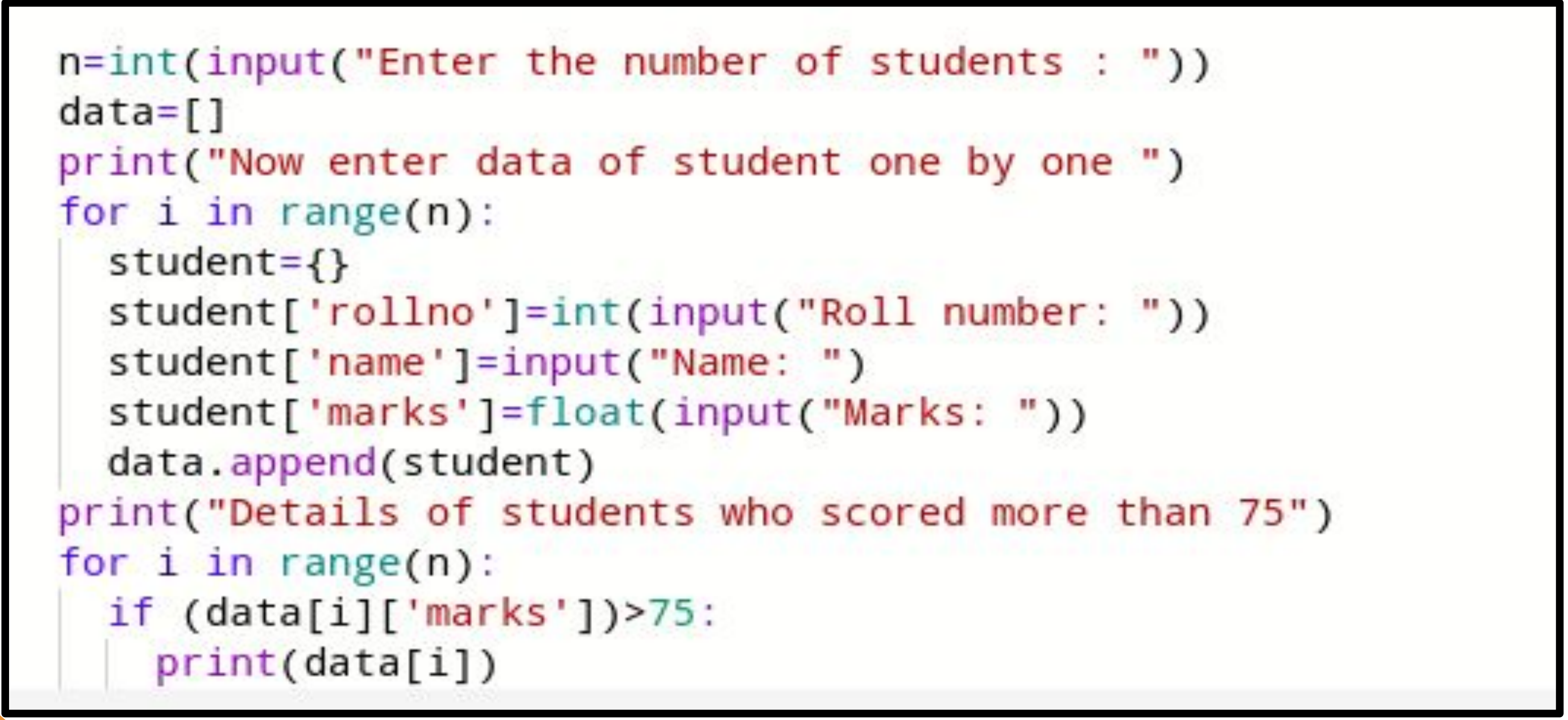
OUTPUT



```
Enter the number of employees : 3
Now enter data of employee one by one
Employee number : 1
Enter the name of employee :KHUSHI
Enter the basic salary of employee :8888
Enter the dept of employee :IT
Employee number : 2
Enter the name of employee :ARUNA
Enter the basic salary of employee :6665
Enter the dept of employee :FINANCE
Employee number : 3
Enter the name of employee :RAJ
Enter the basic salary of employee :9999
Enter the dept of employee :SALES
DATA OF EMPLOYEE  1 {'name': 'KHUSHI', 'basic': 8888.0, 'dept': 'IT', 'hra': 3555.2000000000003,
 'da': 4444.0, 'ta': 888.8000000000001, 'gross': 17776.0, 'pf': 755.48, 'net': 17020.52}
DATA OF EMPLOYEE  2 {'name': 'ARUNA', 'basic': 6665.0, 'dept': 'FINANCE', 'hra': 2666.0, 'da': 3
332.5, 'ta': 666.5, 'gross': 13330.0, 'pf': 566.5250000000001, 'net': 12763.475}
DATA OF EMPLOYEE  3 {'name': 'RAJ', 'basic': 9999.0, 'dept': 'SALES', 'hra': 3999.6000000000004,
 'da': 4999.5, 'ta': 999.9000000000001, 'gross': 19998.0, 'pf': 849.9150000000001, 'net': 19148.
085}
```

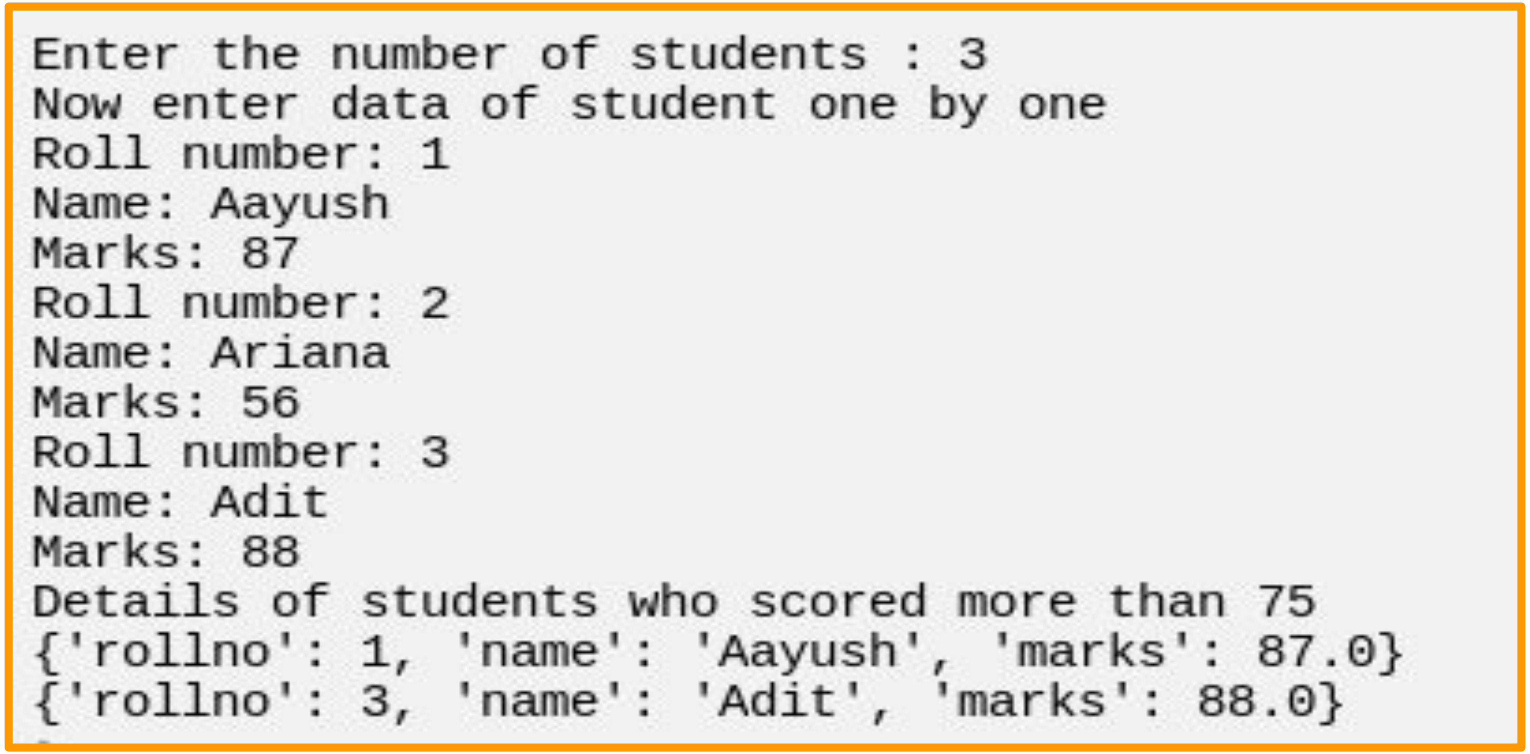
Program #3

To input the roll numbers, names, and marks of n students of a class .Write a Python program to display list of all the students having marks more than 75.



```python
n=int(input("Enter the number of students : "))
data=[]
print("Now enter data of student one by one ")
for i in range(n):
  student={}
  student['rollno']=int(input("Roll number: "))
  student['name']=input("Name: ")
  student['marks']=float(input("Marks: "))
  data.append(student)
print("Details of students who scored more than 75")
for i in range(n):
  if (data[i]['marks'])>75:
    print(data[i])
```

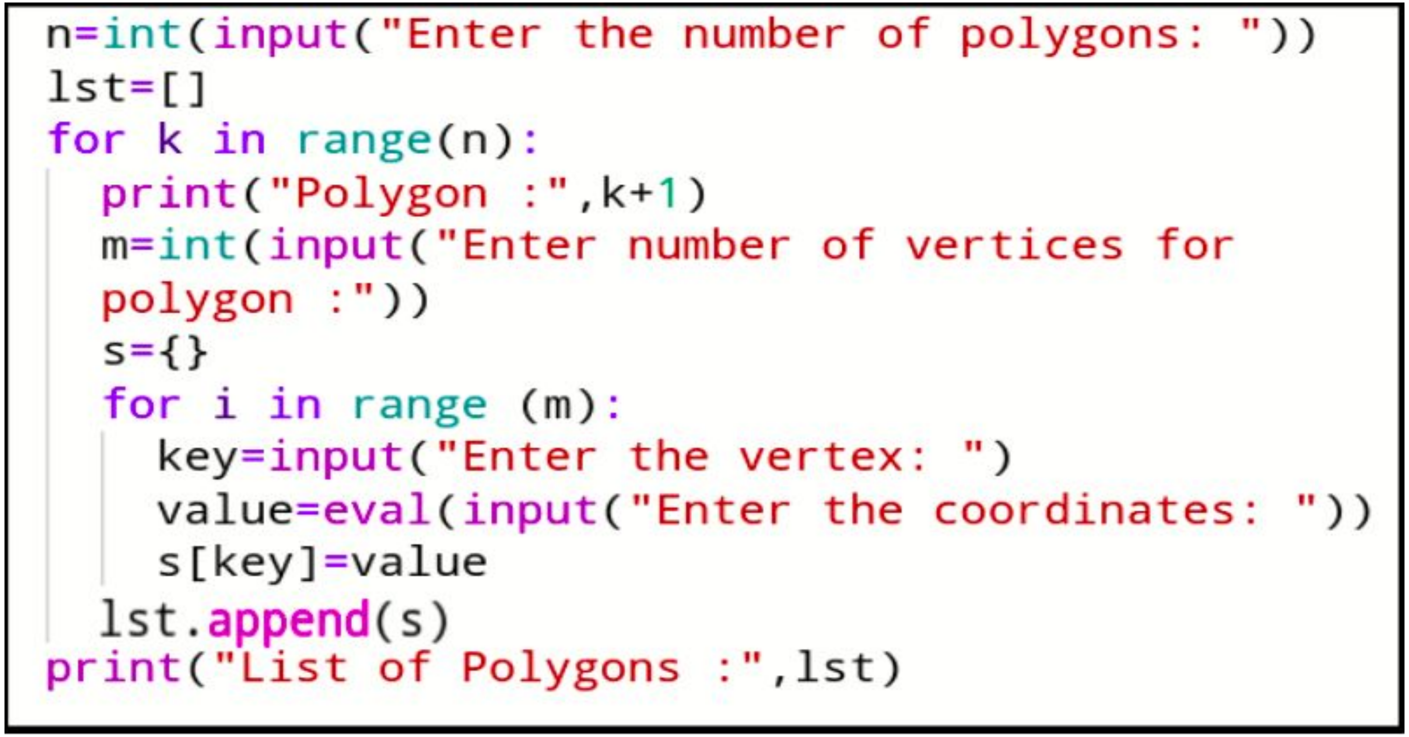
Solution



```
Enter the number of students : 3
Now enter data of student one by one
Roll number: 1
Name: Aayush
Marks: 87
Roll number: 2
Name: Ariana
Marks: 56
Roll number: 3
Name: Adit
Marks: 88
Details of students who scored more than 75
{'rollno': 1, 'name': 'Aayush', 'marks': 87.0}
{'rollno': 3, 'name': 'Adit', 'marks': 88.0}
```

Program #4

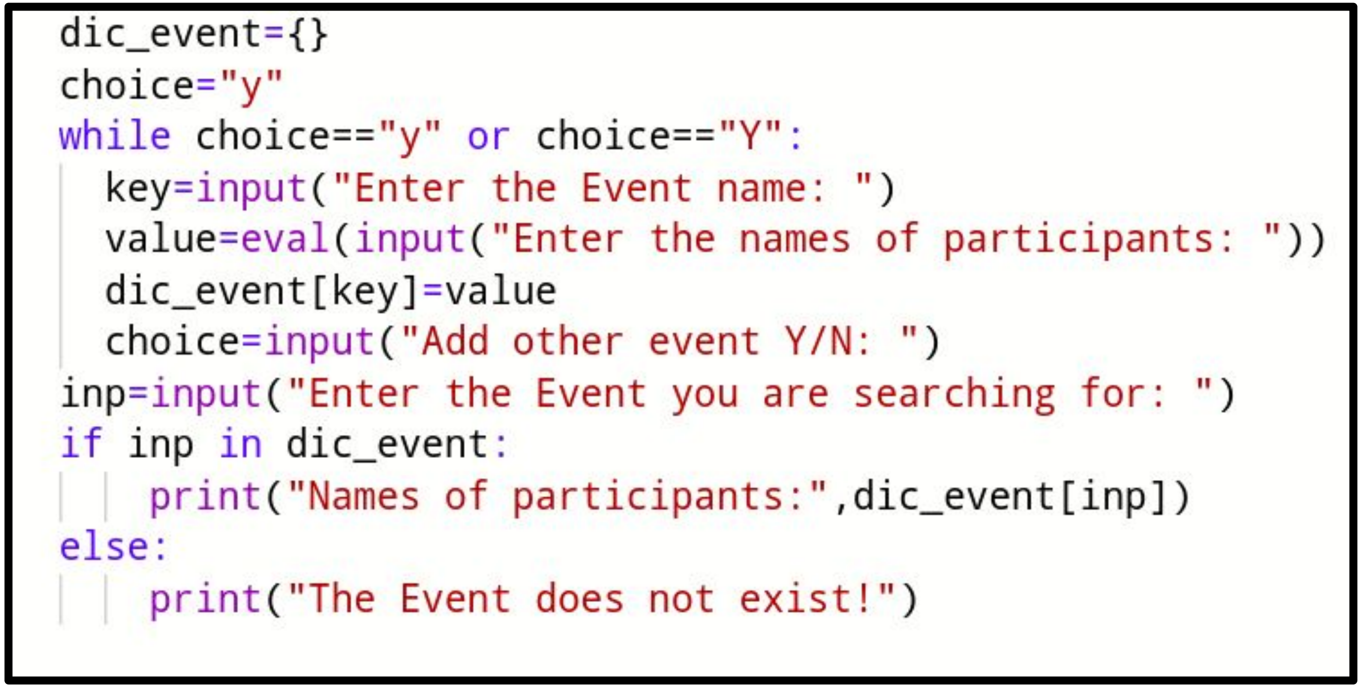
To create a list of polygons, where each polygon is a dictionary storing vertex names and their coordinates as key : value pairs. Then display the coordinates of all the vertices of polygons from this list.



```
n=int(input("Enter the number of polygons: "))
lst=[]
for k in range(n):
  print("Polygon :",k+1)
  m=int(input("Enter number of vertices for
  polygon :"))
  s={}
  for i in range (m):
    key=input("Enter the vertex: ")
    value=eval(input("Enter the coordinates: "))
    s[key]=value
  lst.append(s)
print("List of Polygons :",lst)
```

Program #5

On the occasion of an annual festival, a school is organizing different events. Some of these events are solo and some are team events. Write a program to create a dictionary named Event in which each key is an event name and its value is a list of participants in that event. Then input an event name and display the names of participants of that event.



```python
dic_event={}
choice="y"
while choice=="y" or choice=="Y":
    key=input("Enter the Event name: ")
    value=eval(input("Enter the names of participants: "))
    dic_event[key]=value
    choice=input("Add other event Y/N: ")
inp=input("Enter the Event you are searching for: ")
if inp in dic_event:
        print("Names of participants:",dic_event[inp])
else:
        print("The Event does not exist!")
```

PROGRAM :

To create a dictionary PRODUCT with the keys:

a) productno: Product Number
b) prodname: name of product
c) quantity: quantity of product
d) price: price of the product

Then add the following keys to the dictionary:

e)amount : multiplication of quantity and price

Now enter details of n products.

Make menu based program:

1) To display all details of various products.
2) To search for particular product details for given productname/productno
3) To remove entire detail of one particular product
4) To add a new entry of the product in existing list of details

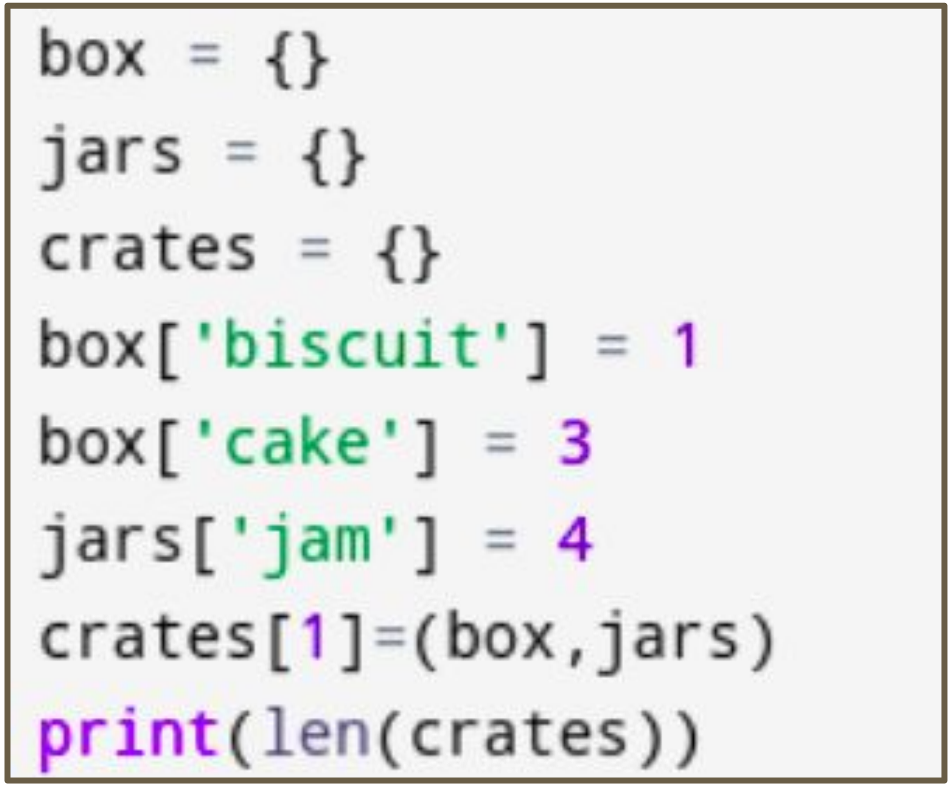
Solution :

```
[illegible] "To display all details of various products\n [illegible] productname/productno\n
    [illegible] details\n
    elif [illegible] ("Enter the productno to be searched :"))
        for i in [illegible] values():
            [illegible] (i)
                break
```

Solution :

```
elif [illegible] ("Enter the productno to be deleted :"))
elif [illegible] ("Enter the productname :"))
ans= [illegible] you want to continue :")
```

Find the output :

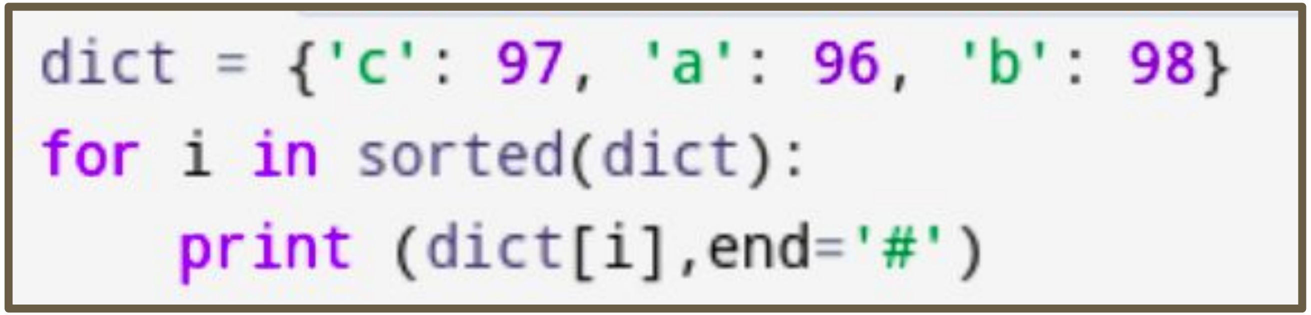


```
box = {}
jars = {}
crates = {}
box['biscuit'] = 1
box['cake'] = 3
jars['jam'] = 4
crates[1]=(box,jars)
print(len(crates))
```

Solution

```
1
```

Find the output :

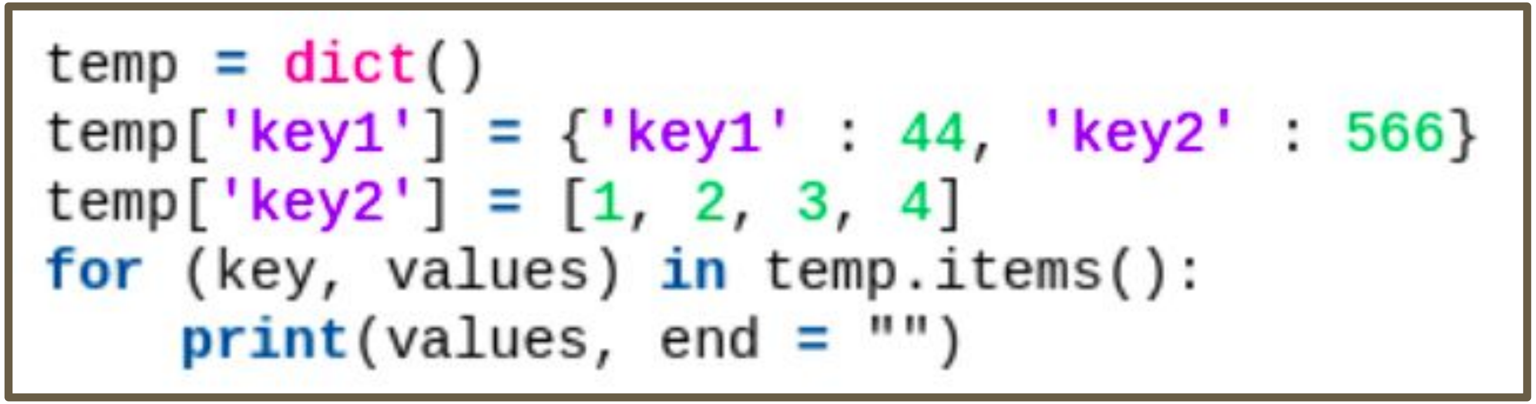


```
dict = {'c': 97, 'a': 96, 'b': 98}
for i in sorted(dict):
    print (dict[i],end='#')
```

Solution

```
96#98#97#
```

Find the output :

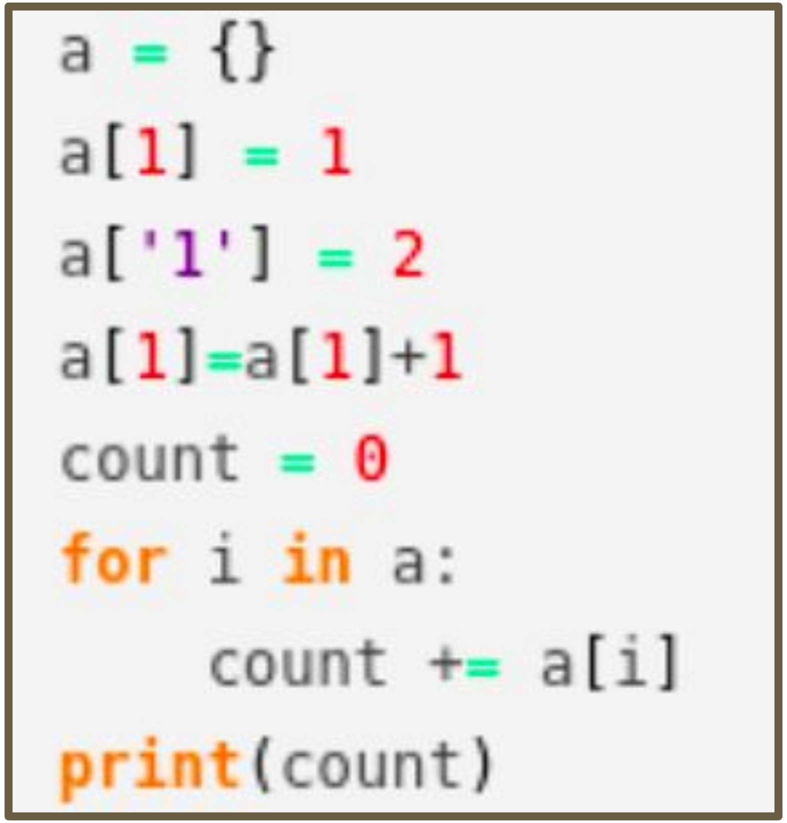


```python
temp = dict()
temp['key1'] = {'key1' : 44, 'key2' : 566}
temp['key2'] = [1, 2, 3, 4]
for (key, values) in temp.items():
    print(values, end = "")
```

Solution

```
{'key1': 44, 'key2': 566} [1, 2, 3, 4]
```

Find the output :



```
a = {}
a[1] = 1
a['1'] = 2
a[1]=a[1]+1
count = 0
for i in a:
    count += a[i]
print(count)
```

Solution

```
4
```

Find the output :

```
D1={1:40,2:60,3:90};D2={3:70}
for k in D1:
    if k in D2 :
        D1[k] -= D2[k]
for k in D1:
    print(D1[k],end='$')
```

Solution

```
40$60$20$
```

Find the output :

```
D={10:"A",20:"B",30:"C"}; L=[30,20,10]
for k in L:
    print(k,D.pop(k),end='#')
```

Solution

```
30 C#20 B#10 A#
```