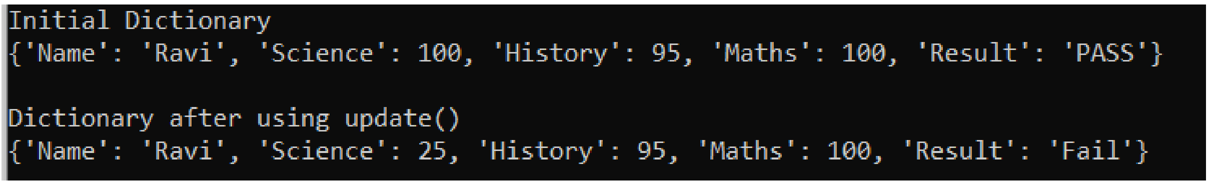
Python – Dictionary and its Operations (Continuation)

Update Dictionary

The `update()` method will update the dictionary with the items from a given argument. If the item does not exist, the item will be added.

The argument must be a dictionary, or an iterable object with key:value pairs.

```
student_dict = {
  "Name": "Ravi",
  "Science": 100,
  "History": 95,
  "Maths": 100,
  "Result":"PASS"
}

print("\nInitial Dictionary")
print(student_dict)

student_dict.update({"Science": 25})
student_dict.update({"Result": "Fail"})


print("\nDictionary after using update()")
print(student_dict)
```



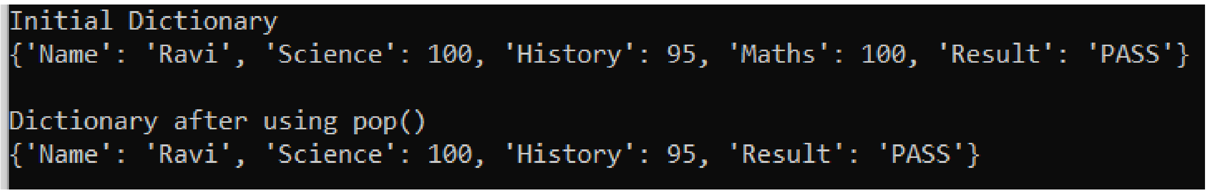
```
Initial Dictionary
{'Name': 'Ravi', 'Science': 100, 'History': 95, 'Maths': 100, 'Result': 'PASS'}

Dictionary after using update()
{'Name': 'Ravi', 'Science': 25, 'History': 95, 'Maths': 100, 'Result': 'Fail'}
```

Removing Items

There are several methods to remove items from a dictionary:

The pop() method removes the item with the specified key name:

```
student_dict = {
  "Name": "Ravi",
  "Science": 100,
  "History": 95,
  "Maths": 100,
  "Result":"PASS"
}

print("\nInitial Dictionary")
print(student_dict)


student_dict.pop("Maths")

print("\nDictionary after using pop()")
print(student_dict)
```

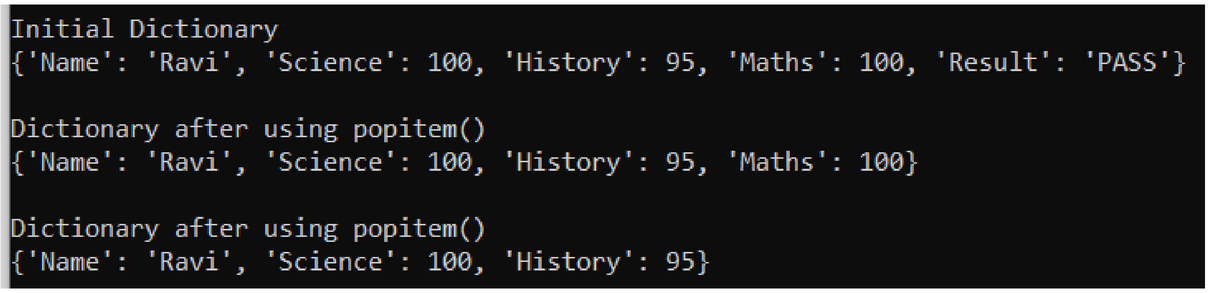


```
Initial Dictionary
{'Name': 'Ravi', 'Science': 100, 'History': 95, 'Maths': 100, 'Result': 'PASS'}

Dictionary after using pop()
{'Name': 'Ravi', 'Science': 100, 'History': 95, 'Result': 'PASS'}
```

The popitem() method removes the last inserted item (in versions before 3.7, a random item is removed instead):

```
student_dict = {
  "Name": "Ravi",
  "Science": 100,
  "History": 95,
  "Maths": 100,
  "Result":"PASS"
}

print("\nInitial Dictionary")
print(student_dict)

student_dict.popitem()
```

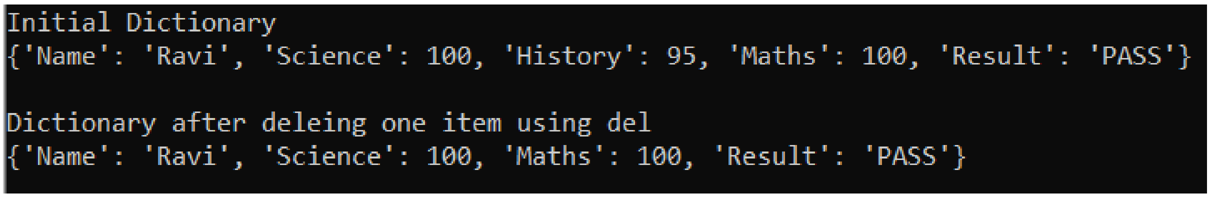
```
print("\nDictionary after using popitem()")
print(student_dict)

student_dict.popitem()

print("\nDictionary after using popitem()")
print(student_dict)
```



```
Initial Dictionary
{'Name': 'Ravi', 'Science': 100, 'History': 95, 'Maths': 100, 'Result': 'PASS'}

Dictionary after using popitem()
{'Name': 'Ravi', 'Science': 100, 'History': 95, 'Maths': 100}

Dictionary after using popitem()
{'Name': 'Ravi', 'Science': 100, 'History': 95}
```

The del keyword removes the item with the specified key name:

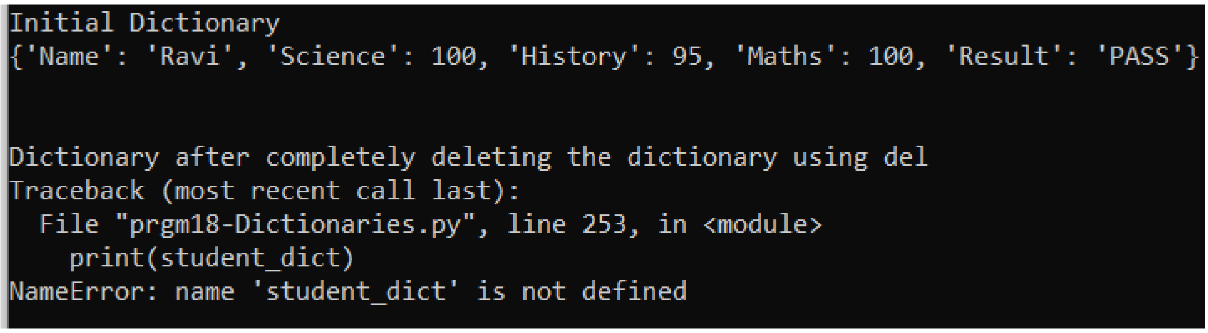
```
student_dict = {
  "Name": "Ravi",
  "Science": 100,
  "History": 95,
  "Maths": 100,
  "Result":"PASS"
}

print("\nInitial Dictionary")
print(student_dict)


del student_dict["History"]

print("\nDictionary after deleing one item using del")
print(student_dict)
```



```
Initial Dictionary
{'Name': 'Ravi', 'Science': 100, 'History': 95, 'Maths': 100, 'Result': 'PASS'}

Dictionary after deleing one item using del
{'Name': 'Ravi', 'Science': 100, 'Maths': 100, 'Result': 'PASS'}
```

The del keyword can also delete the dictionary completely:

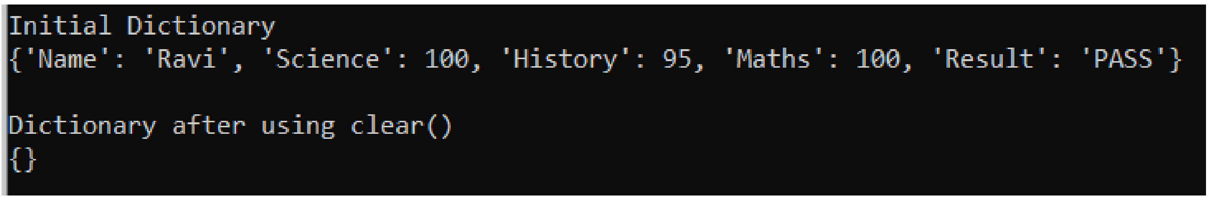
```
student_dict = {
  "Name": "Ravi",
  "Science": 100,
  "History": 95,
  "Maths": 100,
  "Result":"PASS"
}

print("\nInitial Dictionary")
print(student_dict)

del student_dict

print("\n\nDictionary after completely deleting the dictionary using del")
print(student_dict) #this will cause an error because "student_dict" no longer exists.
```



```
Initial Dictionary
{'Name': 'Ravi', 'Science': 100, 'History': 95, 'Maths': 100, 'Result': 'PASS'}


Dictionary after completely deleting the dictionary using del
Traceback (most recent call last):
  File "prgm18-Dictionaries.py", line 253, in <module>
    print(student_dict)
NameError: name 'student_dict' is not defined
```

The clear() method empties the dictionary:

```
student_dict = {
  "Name": "Ravi",
  "Science": 100,
  "History": 95,
  "Maths": 100,
  "Result":"PASS"
}
print("\nInitial Dictionary")
print(student_dict)

student_dict.clear()

print("\nDictionary after using clear()")
print(student_dict)
```

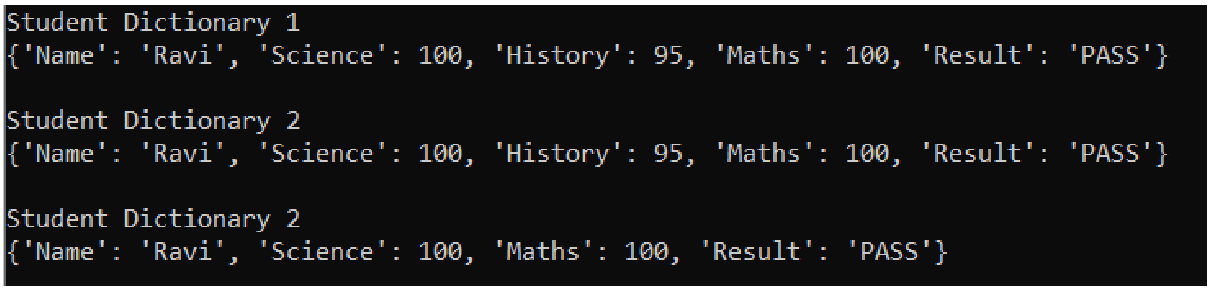


```
Initial Dictionary
{'Name': 'Ravi', 'Science': 100, 'History': 95, 'Maths': 100, 'Result': 'PASS'}

Dictionary after using clear()
{}
```

Copy a Dictionary

You cannot copy a dictionary simply by typing `dict2 = dict1`, because: `dict2` will only be a *reference* to `dict1`, and changes made in `dict1` will automatically also be made in `dict2`.

```
student_dict1 = {
  "Name": "Ravi",
  "Science": 100,
  "History": 95,
  "Maths": 100,
  "Result":"PASS"
}

print("\nStudent Dictionary 1")
print(student_dict1)


student_dict2=student_dict1

print("\nStudent Dictionary 2")
print(student_dict2)


del student_dict1["History"]

print("\nStudent Dictionary 2")
print(student_dict2)
```

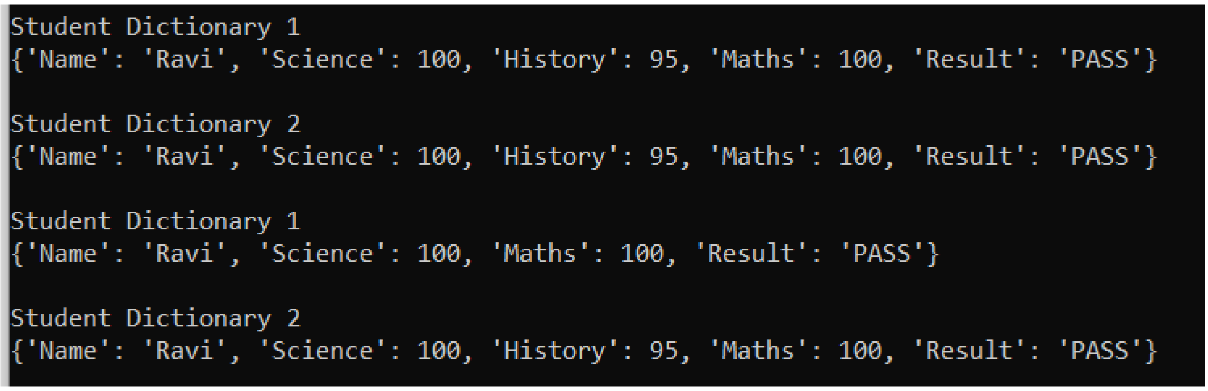


```
Student Dictionary 1
{'Name': 'Ravi', 'Science': 100, 'History': 95, 'Maths': 100, 'Result': 'PASS'}

Student Dictionary 2
{'Name': 'Ravi', 'Science': 100, 'History': 95, 'Maths': 100, 'Result': 'PASS'}

Student Dictionary 2
{'Name': 'Ravi', 'Science': 100, 'Maths': 100, 'Result': 'PASS'}
```

There are ways to make a copy, one way is to use the built-in Dictionary method `copy()`.

Make a copy of a dictionary with the `copy()` method:

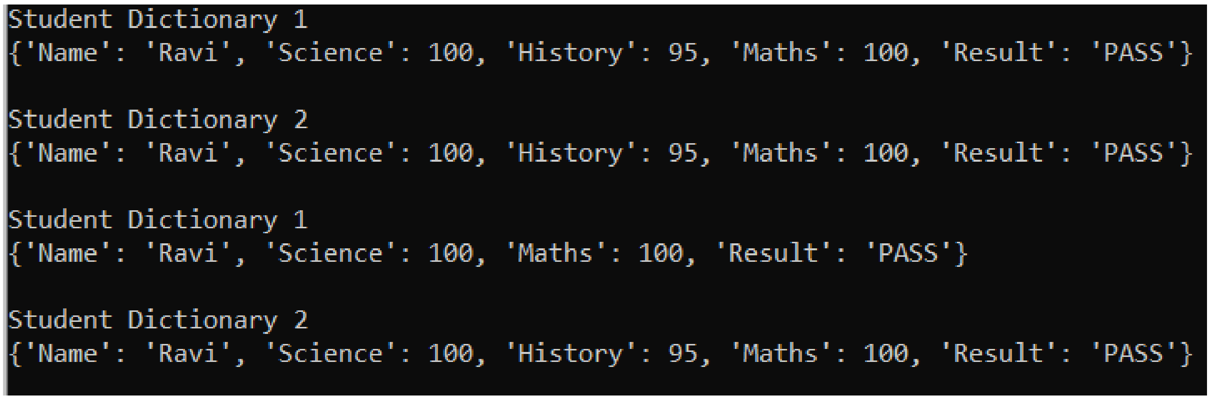
```
student_dict1 = {
  "Name": "Ravi",
  "Science": 100,
  "History": 95,
  "Maths": 100,
  "Result":"PASS"
}

print("\nStudent Dictionary 1")
print(student_dict1)

student_dict2 = student_dict1.copy()

print("\nStudent Dictionary 2")
print(student_dict2)


del student_dict1["History"]


print("\nStudent Dictionary 1")
print(student_dict1)

print("\nStudent Dictionary 2")
print(student_dict2)
```



```
Student Dictionary 1
{'Name': 'Ravi', 'Science': 100, 'History': 95, 'Maths': 100, 'Result': 'PASS'}

Student Dictionary 2
{'Name': 'Ravi', 'Science': 100, 'History': 95, 'Maths': 100, 'Result': 'PASS'}

Student Dictionary 1
{'Name': 'Ravi', 'Science': 100, 'Maths': 100, 'Result': 'PASS'}

Student Dictionary 2
{'Name': 'Ravi', 'Science': 100, 'History': 95, 'Maths': 100, 'Result': 'PASS'}
```

Another way to make a copy is to use the built-in function `dict()`.

```
student_dict1 = {
  "Name": "Ravi",
  "Science": 100,
  "History": 95,
  "Maths": 100,
  "Result":"PASS"
}

print("\nStudent Dictionary 1")
print(student_dict1)

student_dict2 = dict(student_dict1)

print("\nStudent Dictionary 2")
print(student_dict2)

del student_dict1["History"]

print("\nStudent Dictionary 1")
print(student_dict1)

print("\nStudent Dictionary 2")
print(student_dict2)
```

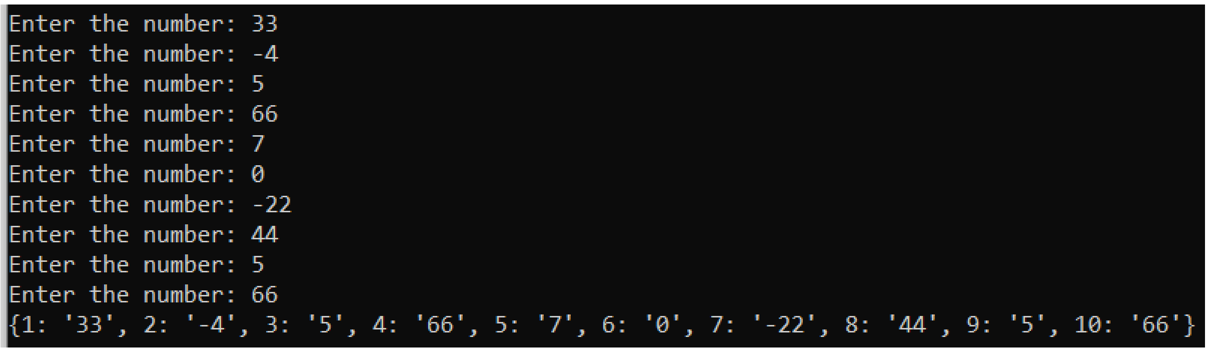


```
Student Dictionary 1
{'Name': 'Ravi', 'Science': 100, 'History': 95, 'Maths': 100, 'Result': 'PASS'}

Student Dictionary 2
{'Name': 'Ravi', 'Science': 100, 'History': 95, 'Maths': 100, 'Result': 'PASS'}

Student Dictionary 1
{'Name': 'Ravi', 'Science': 100, 'Maths': 100, 'Result': 'PASS'}

Student Dictionary 2
{'Name': 'Ravi', 'Science': 100, 'History': 95, 'Maths': 100, 'Result': 'PASS'}
```

Activities using Python Dictionaries

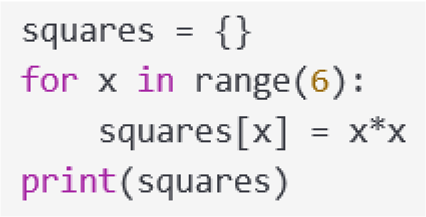
1. Read any 10 numbers (**values**) from user and add them to a dictionary (keys are 1,2, …10). Let the dictionary name be **number_dict**.

   Sample output:



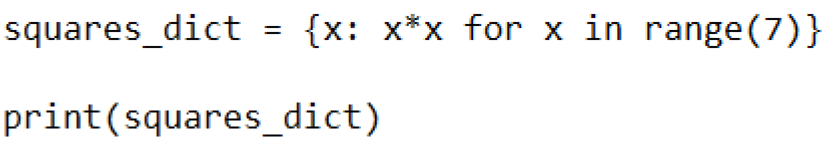
```
Enter the number: 33
Enter the number: -4
Enter the number: 5
Enter the number: 66
Enter the number: 7
Enter the number: 0
Enter the number: -22
Enter the number: 44
Enter the number: 5
Enter the number: 66
{1: '33', 2: '-4', 3: '5', 4: '66', 5: '7', 6: '0', 7: '-22', 8: '44', 9: '5', 10: '66'}
```

2. **Execute the following code and observe the output.**



```
squares = {}
for x in range(6):
    squares[x] = x*x
print(squares)
```

3. **Execute the following code and observe the output.**



```
squares_dict = {x: x*x for x in range(7)}

print(squares_dict)
```