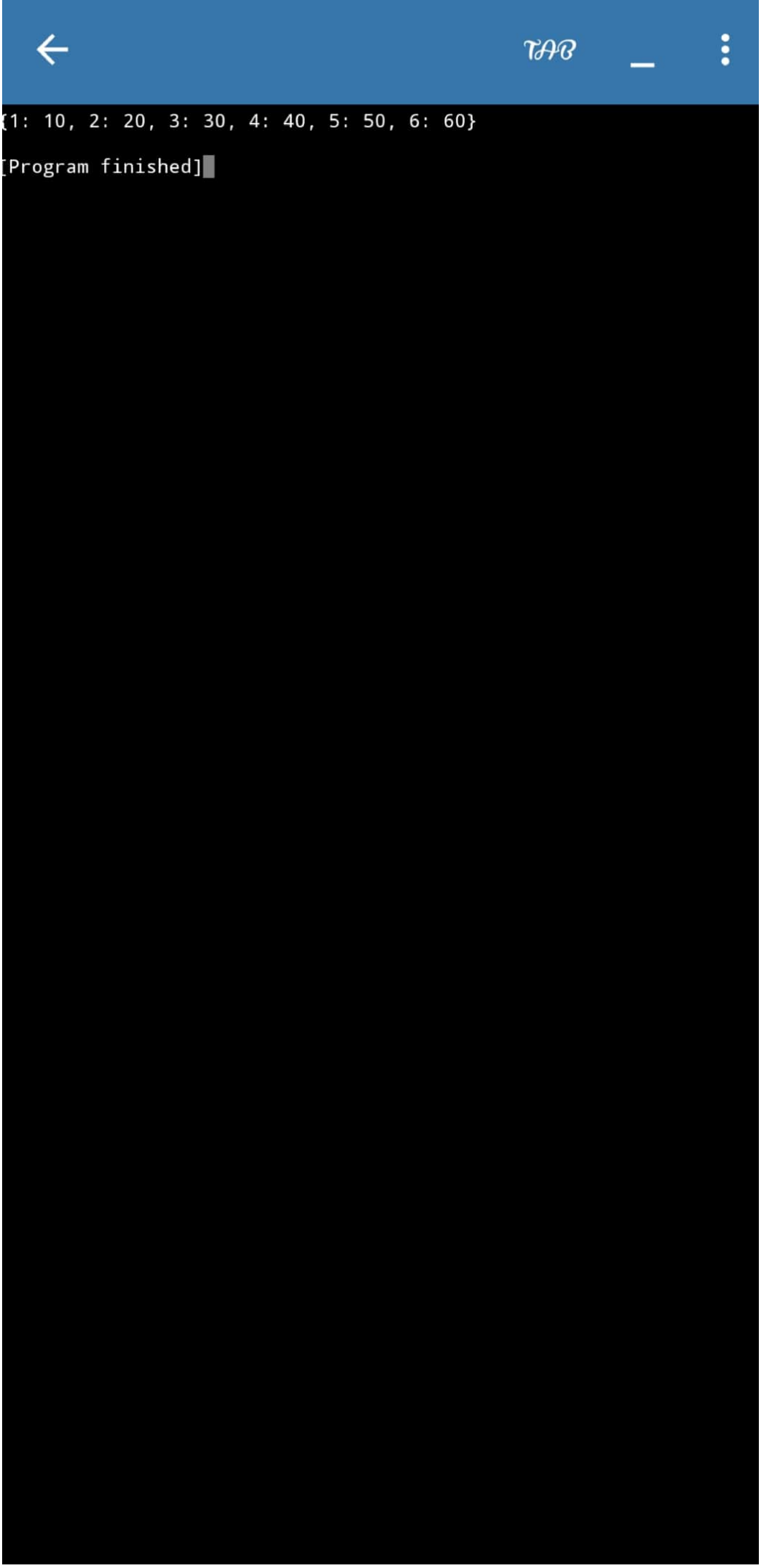
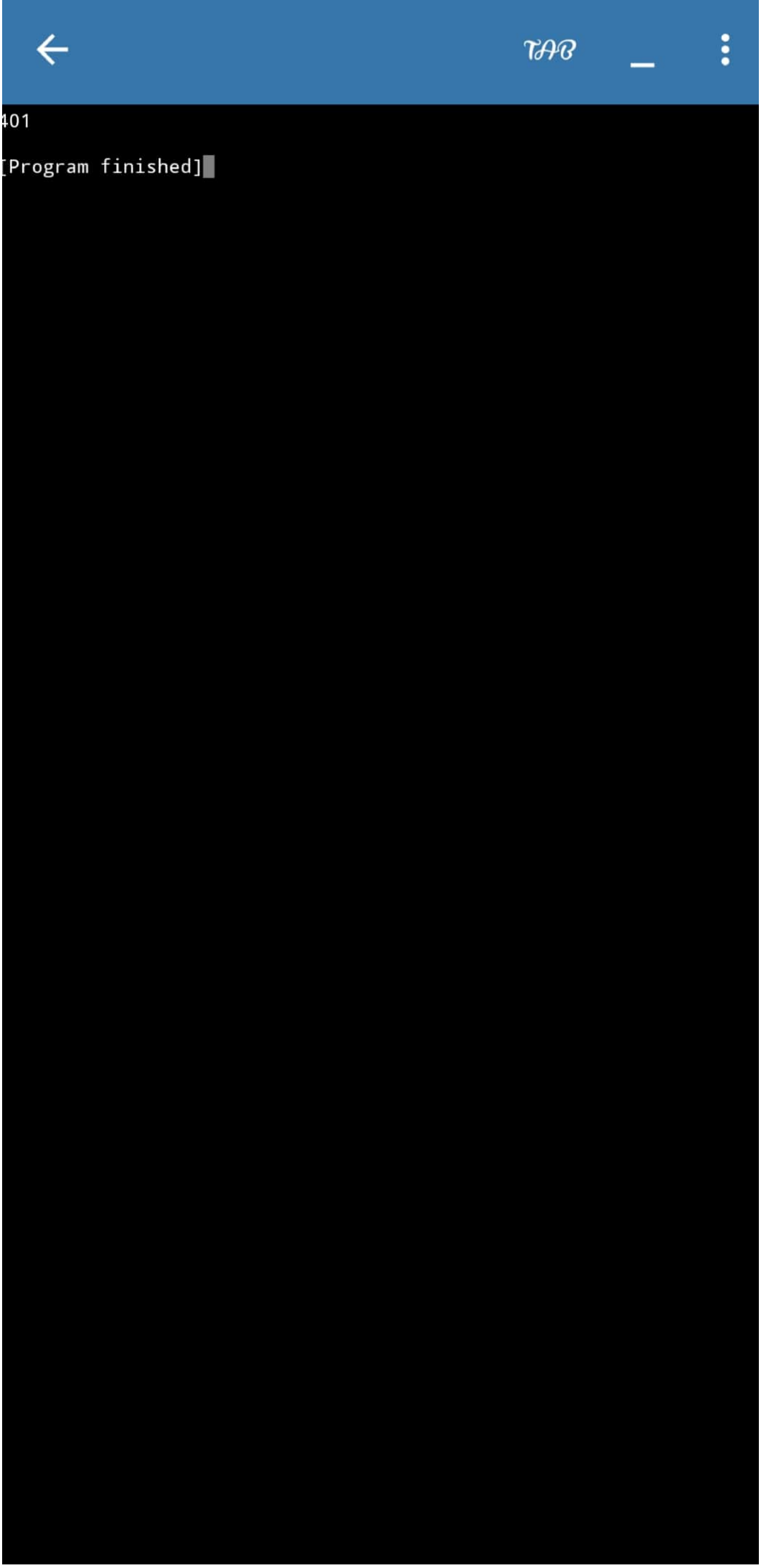
A list can store a sequence of objects in a certain order such that you can index into the list, or iterate over the list. List is a mutable type meaning that lists can be modified after they have been created.
A tuple is similar to a list except it is immutable. There is also a semantic difference between a list and a tuple.
Tuples have structure, lists have order.
A dictionary is a key-value store. It is not ordered and it requires that the keys are hashable. It is fast for lookups by key

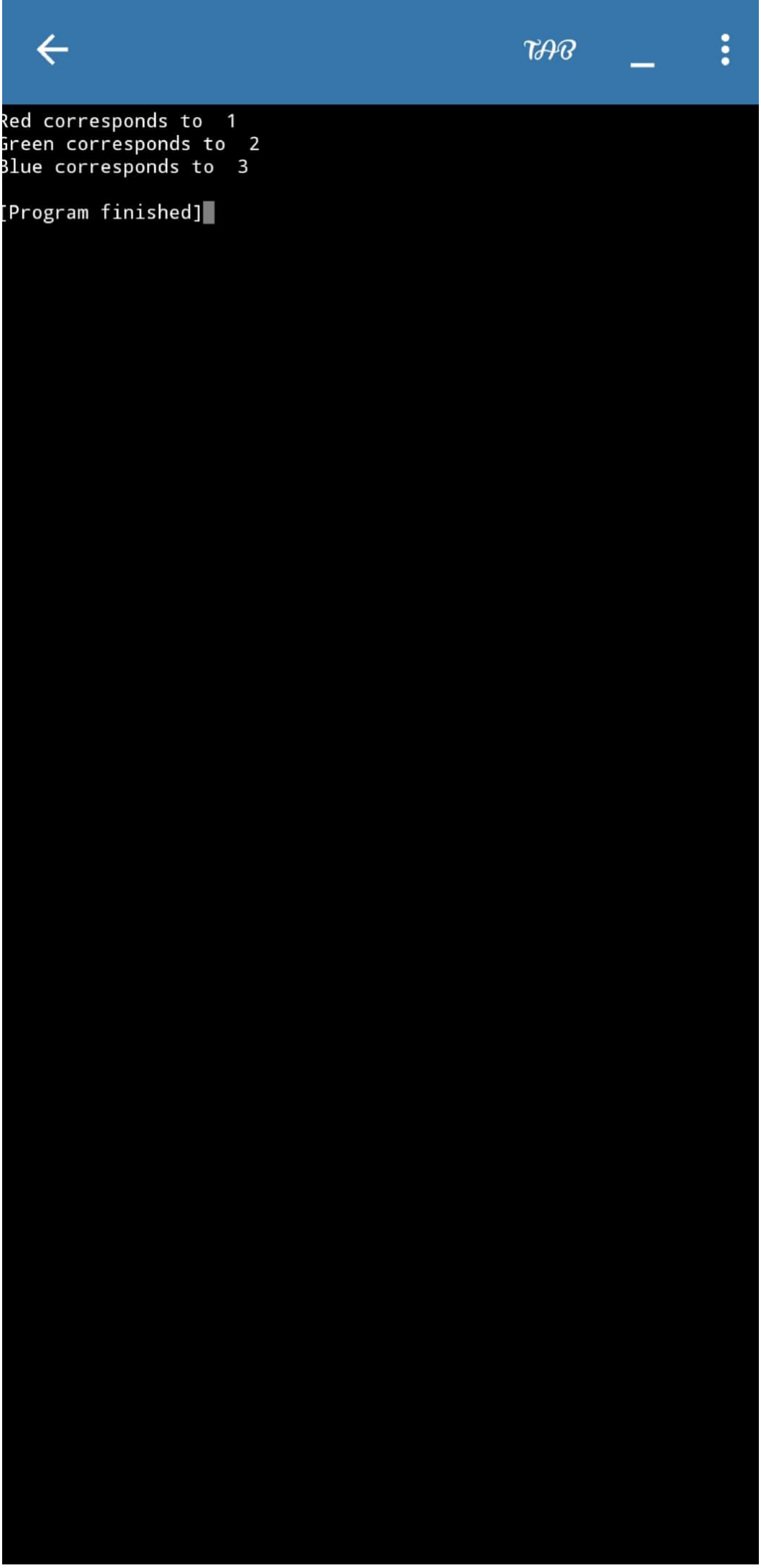
```
1 dic1={1:10, 2:20}
2 dic2={3:30, 4:40}
3 dic3={5:50,6:60}
4 dic4 = {}
5 for d in (dic1, dic2, dic3): dic4.update(d)
6 print(dic4)
```



```
1 my_dict = {100,54,247}
2 print ( sum ( my_dict ) )
3
```



```
1  d = {'Red': 1, 'Green': 2, 'Blue': 3}
2  for color_key, value in d.items( ):
3      print( color_key, 'corresponds to ',
   d[ color_key ] )
4
```



key.py
/storage/emulated/0/Do···

item.py dict.py class.py iterate.py key.py

```
1  d = {1: 10, 2: 20, 3: 30, 4: 40, 5: 50, 6:
   60}
2  def is_key_present(x):
3    if x in d:
4        print('Key is present in the dictionary')
5    else:
6        print('Key is not present in the dictionary')
7  is_key_present(5)
8  is_key_present(9)
9
```