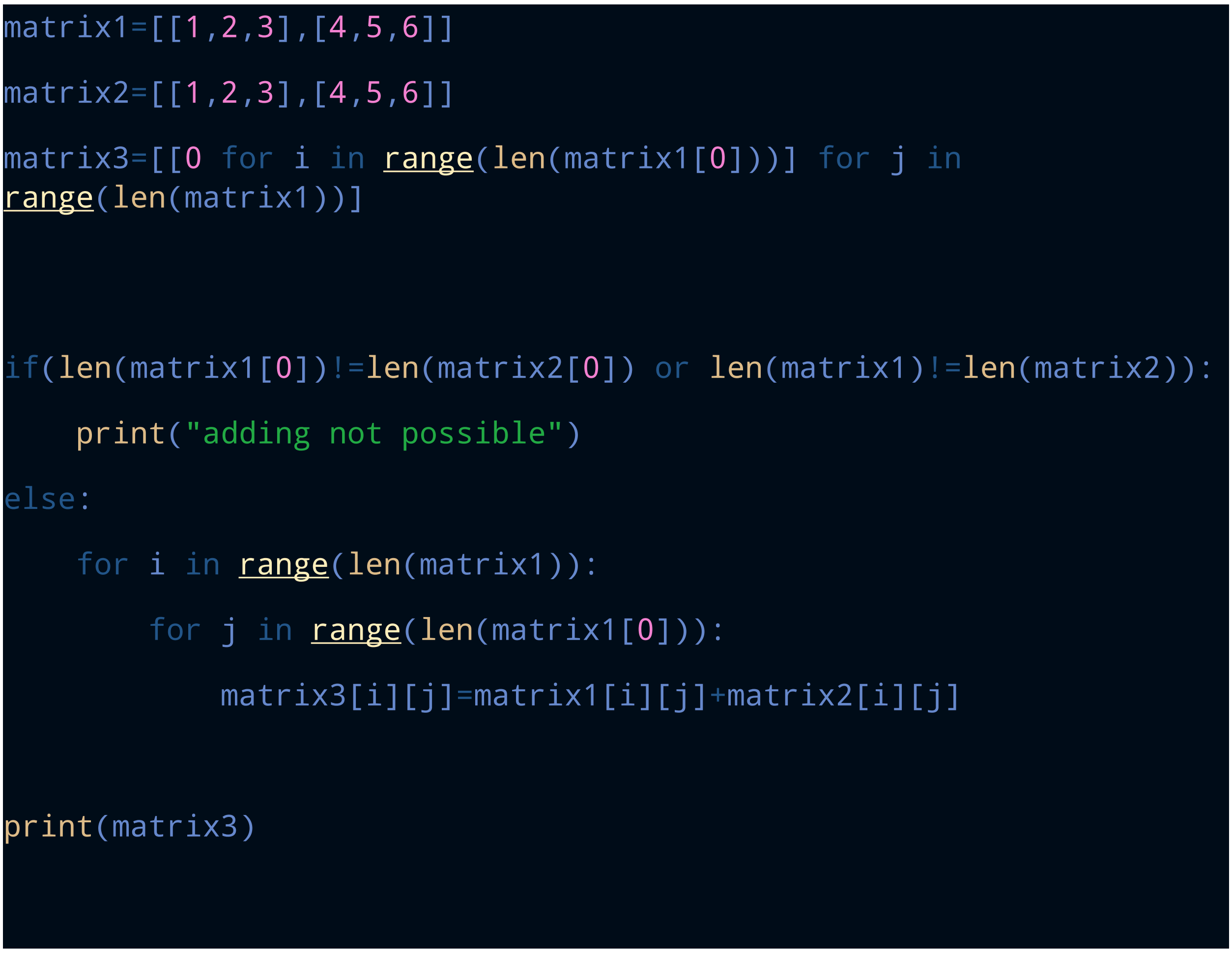
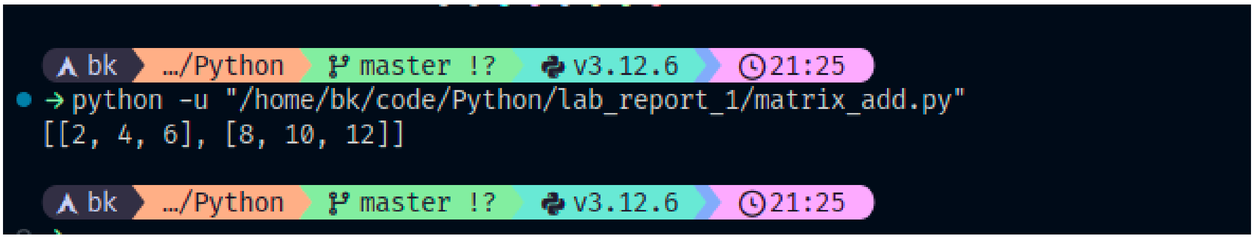
1. Matrix Addition:

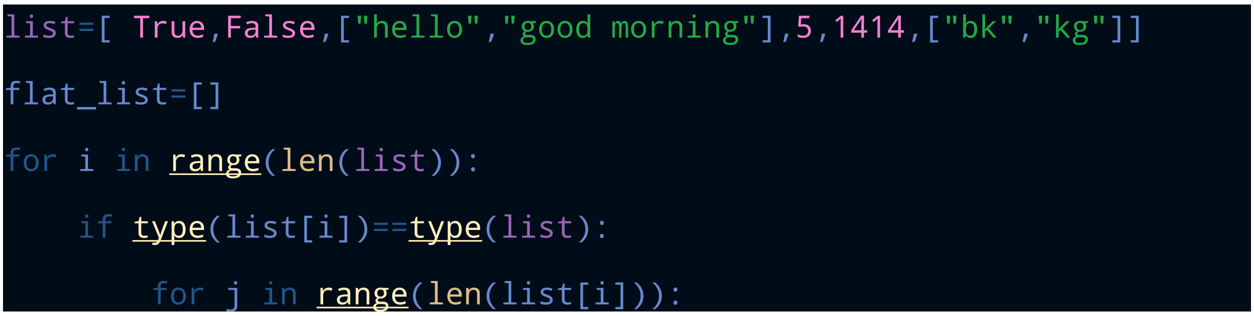


```
matrix1=[[1,2,3],[4,5,6]]

matrix2=[[1,2,3],[4,5,6]]

matrix3=[[0 for i in range(len(matrix1[0]))] for j in
range(len(matrix1))]



if(len(matrix1[0])!=len(matrix2[0]) or len(matrix1)!=len(matrix2)):

    print("adding not possible")

else:

    for i in range(len(matrix1)):

        for j in range(len(matrix1[0])):

            matrix3[i][j]=matrix1[i][j]+matrix2[i][j]


print(matrix3)
```

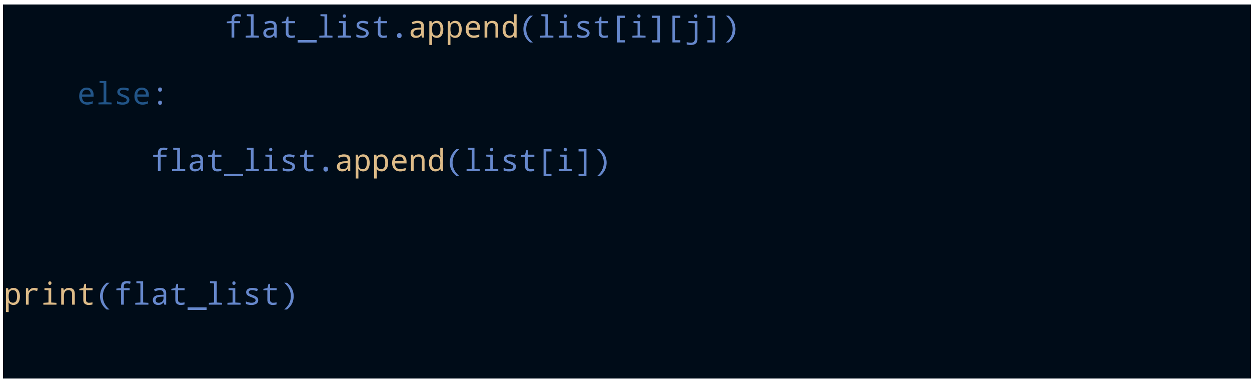


```
 bk  …/Python  master !?  v3.12.6  21:25
● → python -u "/home/bk/code/Python/lab_report_1/matrix_add.py"
 [[2, 4, 6], [8, 10, 12]]

 bk  …/Python  master !?  v3.12.6  21:25
```

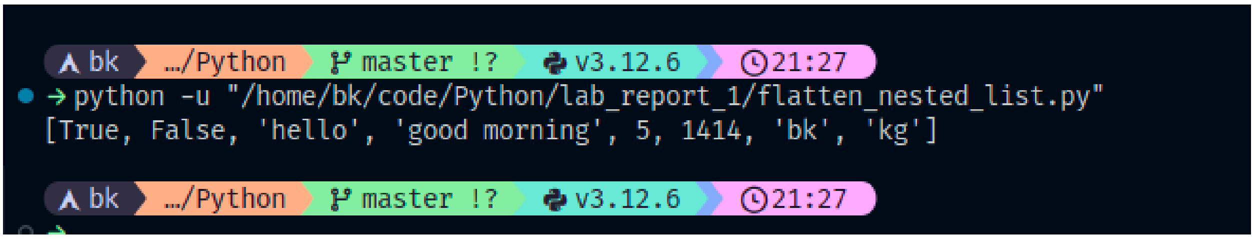
2. Flatten nested loop



```
list=[ True,False,["hello","good morning"],5,1414,["bk","kg"]]

flat_list=[]

for i in range(len(list)):

    if type(list[i])==type(list):

        for j in range(len(list[i])):
```

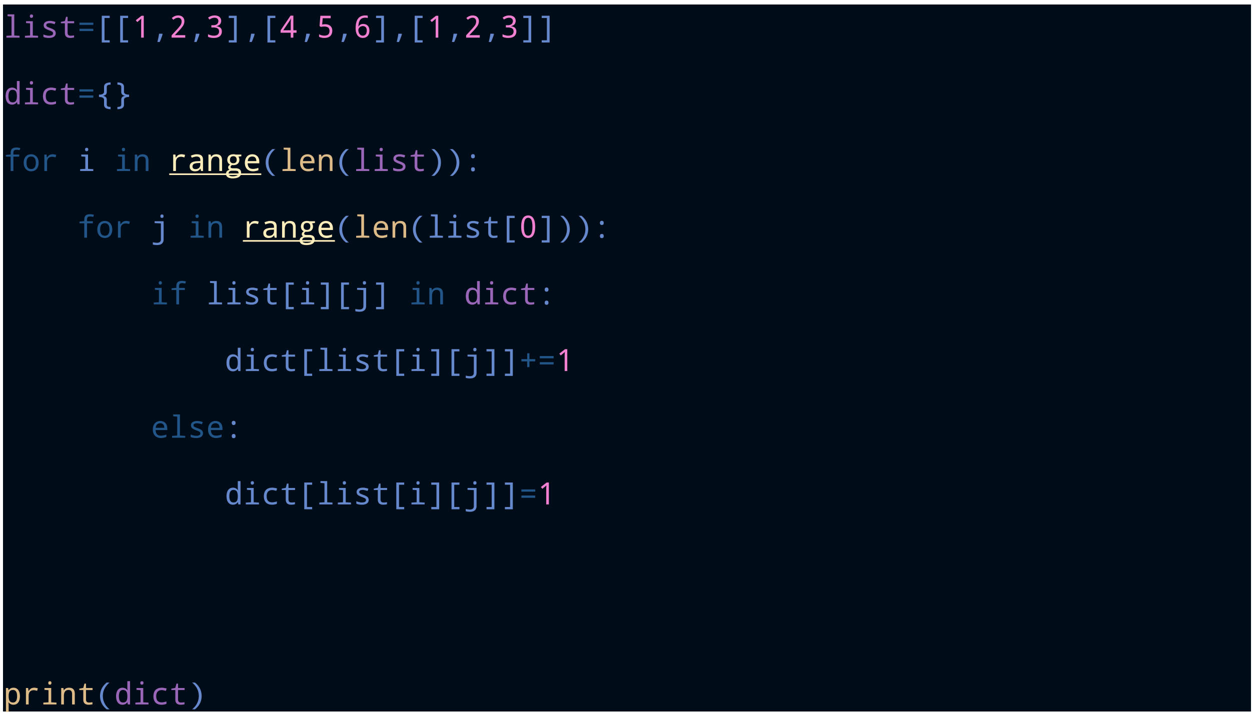


```
            flat_list.append(list[i][j])
    else:
        flat_list.append(list[i])


print(flat_list)
```

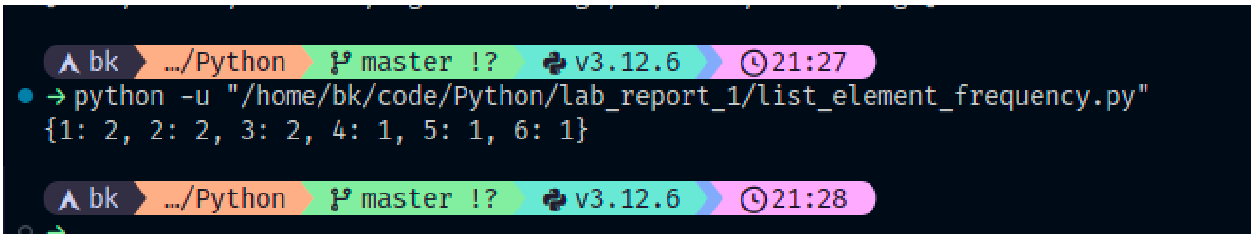


```
 bk  …/Python  master !?  v3.12.6  21:27
● → python -u "/home/bk/code/Python/lab_report_1/flatten_nested_list.py"
[True, False, 'hello', 'good morning', 5, 1414, 'bk', 'kg']

 bk  …/Python  master !?  v3.12.6  21:27
○ →
```

3. List element frequency

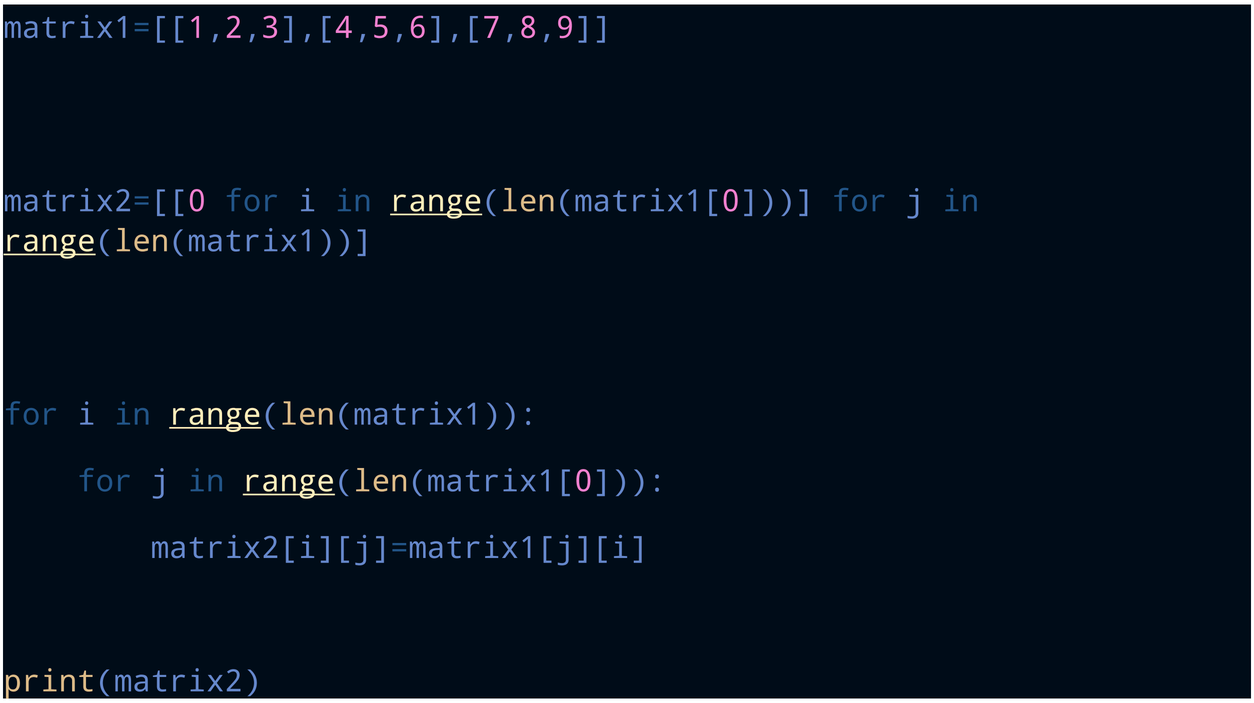


```
list=[[1,2,3],[4,5,6],[1,2,3]]
dict={}
for i in range(len(list)):
    for j in range(len(list[0])):
        if list[i][j] in dict:
            dict[list[i][j]]+=1
        else:
            dict[list[i][j]]=1



print(dict)
```

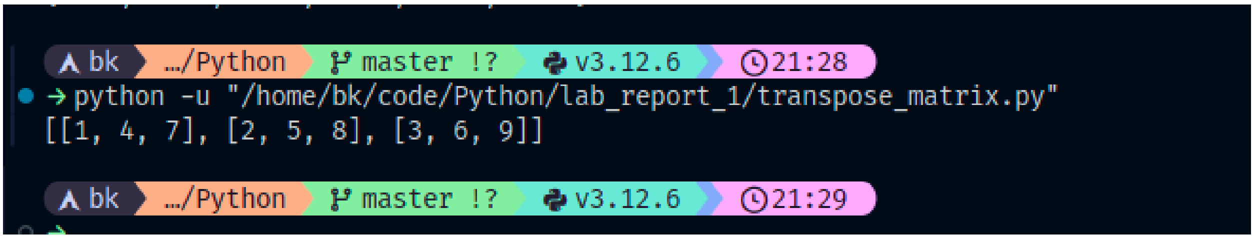


```
 bk  …/Python  master !?  v3.12.6  21:27
● → python -u "/home/bk/code/Python/lab_report_1/list_element_frequency.py"
{1: 2, 2: 2, 3: 2, 4: 1, 5: 1, 6: 1}

 bk  …/Python  master !?  v3.12.6  21:28
○ →
```

4. Transpose matrix

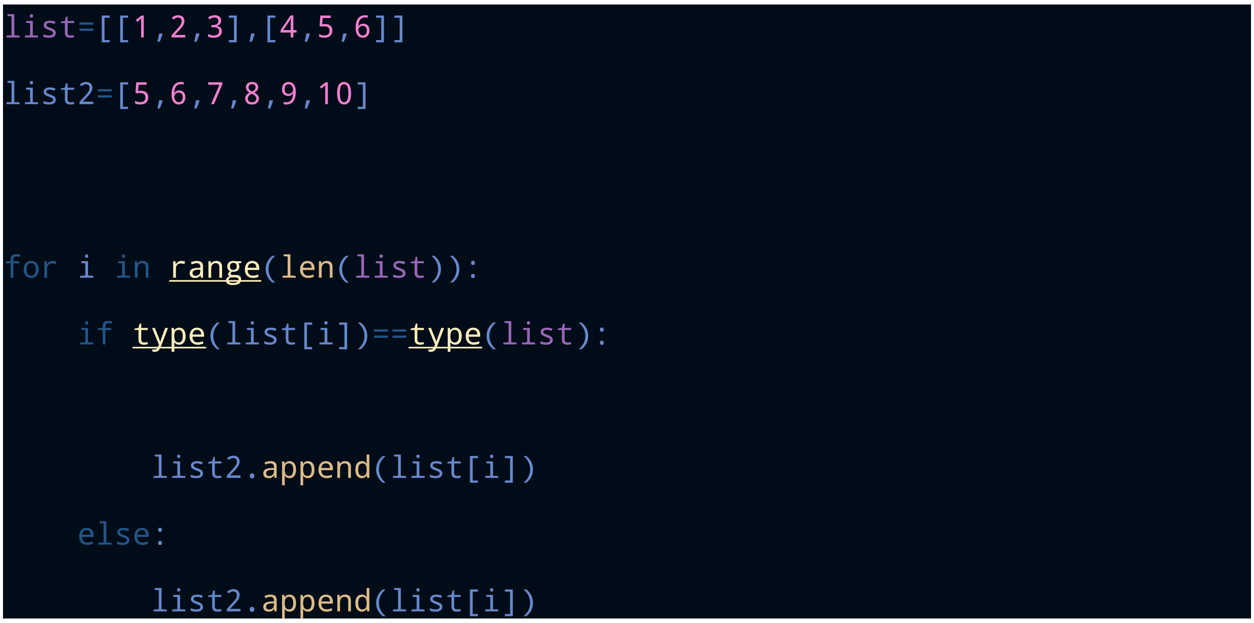


```
matrix1=[[1,2,3],[4,5,6],[7,8,9]]


matrix2=[[0 for i in range(len(matrix1[0]))] for j in
range(len(matrix1))]


for i in range(len(matrix1)):
    for j in range(len(matrix1[0])):
        matrix2[i][j]=matrix1[j][i]

print(matrix2)
```

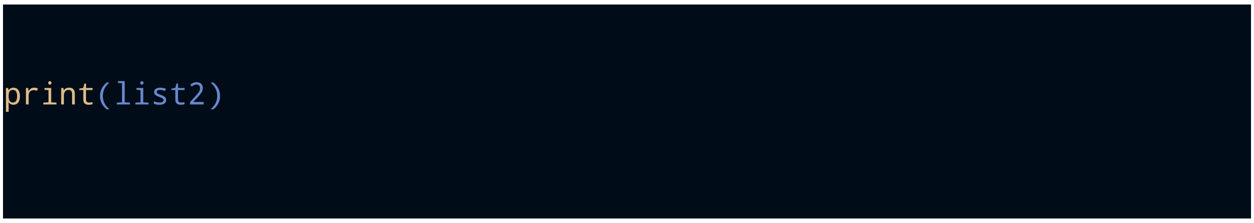


```
 bk  …/Python  master !?  v3.12.6  21:28
→ python -u "/home/bk/code/Python/lab_report_1/transpose_matrix.py"
[[1, 4, 7], [2, 5, 8], [3, 6, 9]]

 bk  …/Python  master !?  v3.12.6  21:29
```

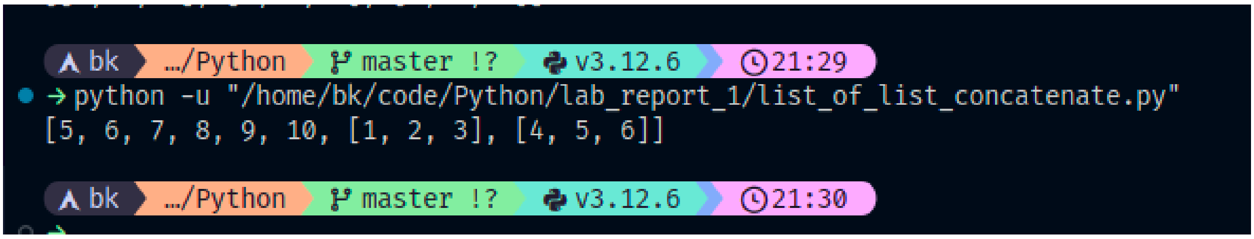
5. List of list concatenation



```
list=[[1,2,3],[4,5,6]]
list2=[5,6,7,8,9,10]


for i in range(len(list)):
    if type(list[i])==type(list):

        list2.append(list[i])
    else:
        list2.append(list[i])
```

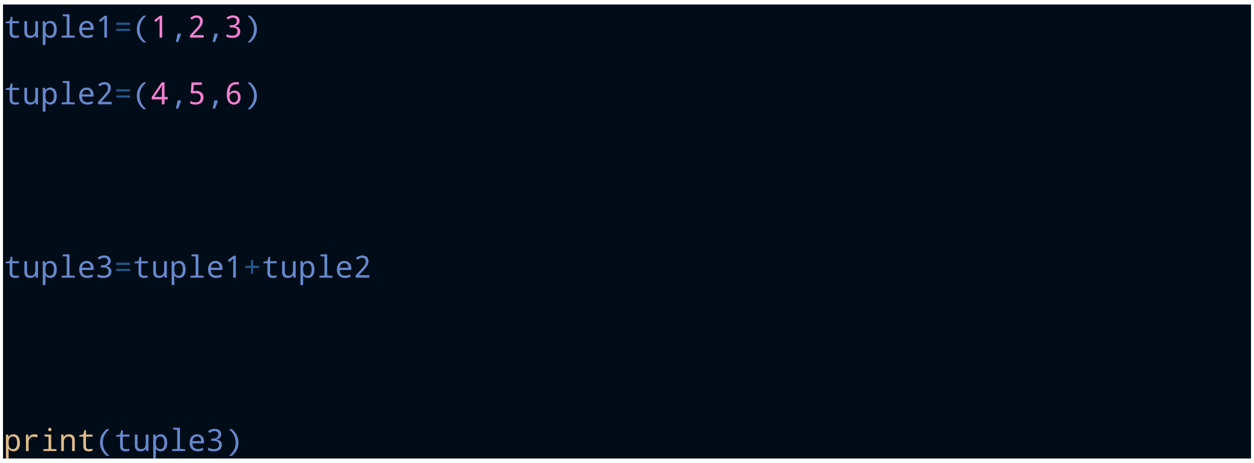


```python
print(list2)
```

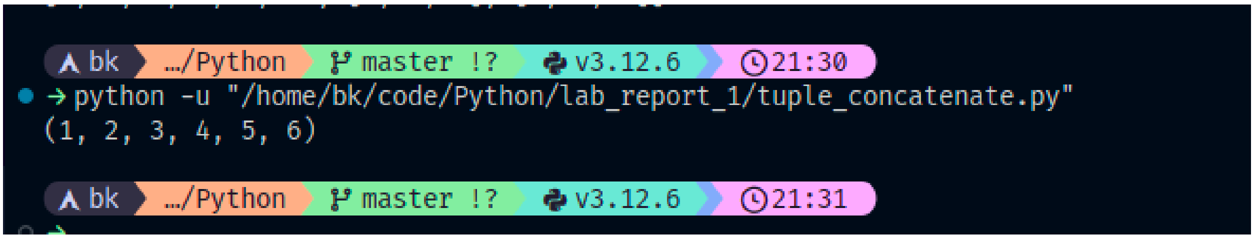


```
 bk  …/Python  master !?  v3.12.6  21:29
● → python -u "/home/bk/code/Python/lab_report_1/list_of_list_concatenate.py"
 [5, 6, 7, 8, 9, 10, [1, 2, 3], [4, 5, 6]]

 bk  …/Python  master !?  v3.12.6  21:30
```

6. Tuple concatenate

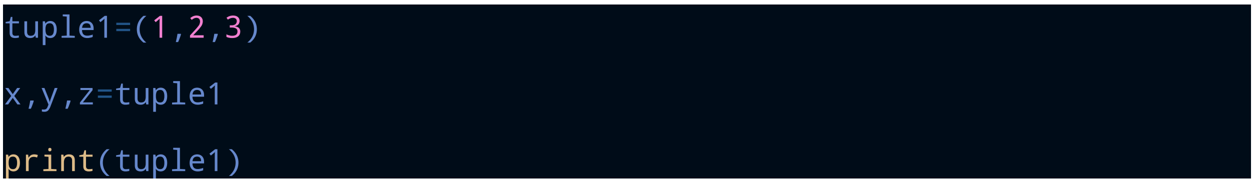


```python
tuple1=(1,2,3)

tuple2=(4,5,6)



tuple3=tuple1+tuple2



print(tuple3)
```

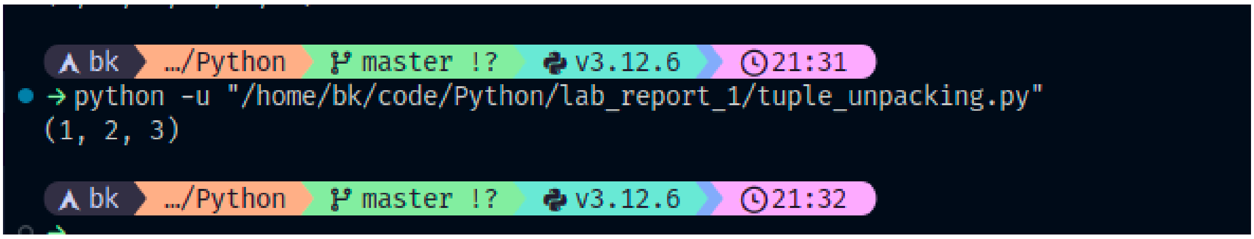


```
 bk  …/Python  master !?  v3.12.6  21:30
● → python -u "/home/bk/code/Python/lab_report_1/tuple_concatenate.py"
 (1, 2, 3, 4, 5, 6)

 bk  …/Python  master !?  v3.12.6  21:31
```

7. Tuple unpacking

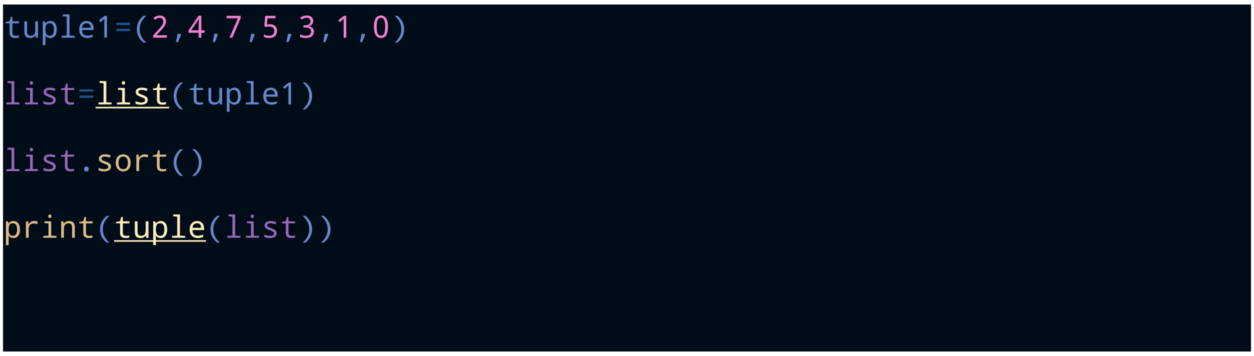


```python
tuple1=(1,2,3)

x,y,z=tuple1

print(tuple1)
```

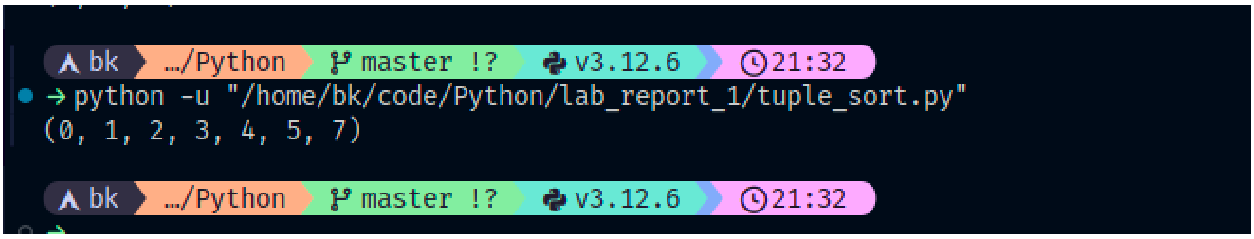


```
 bk  …/Python  master !?  v3.12.6  21:31
● → python -u "/home/bk/code/Python/lab_report_1/tuple_unpacking.py"
(1, 2, 3)

 bk  …/Python  master !?  v3.12.6  21:32
```

8. Tuple sorting

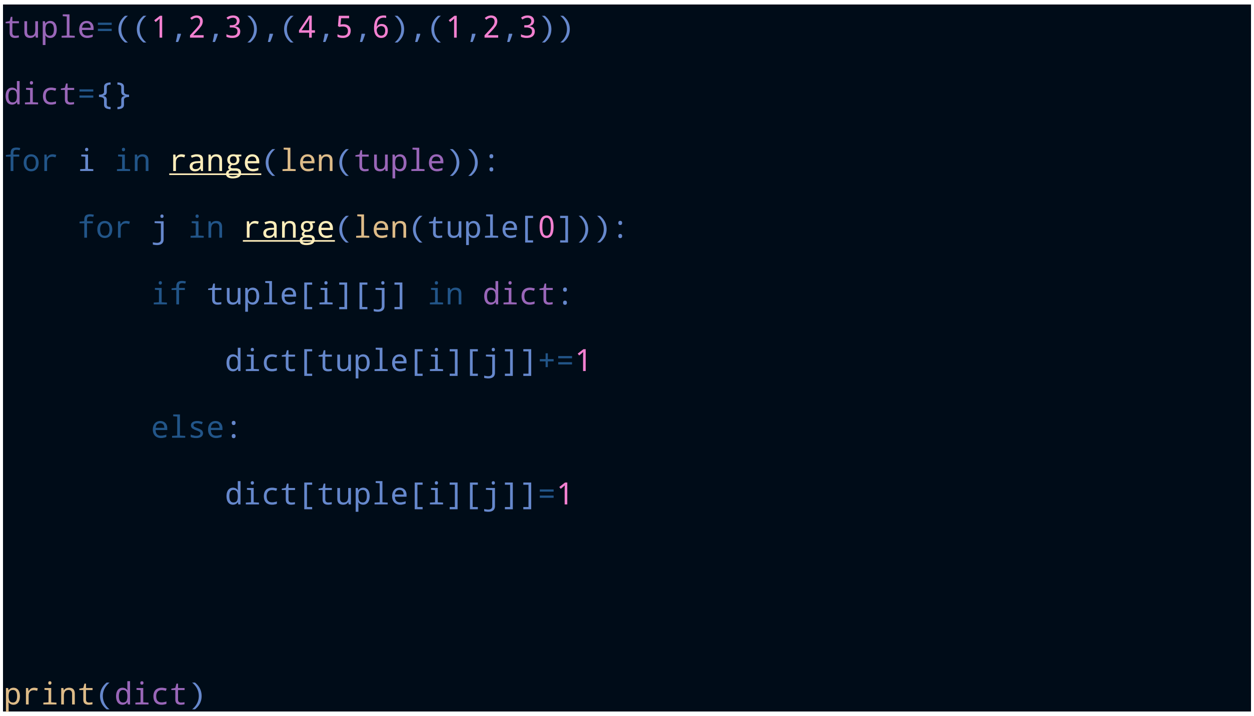


```
tuple1=(2,4,7,5,3,1,0)
list=list(tuple1)
list.sort()
print(tuple(list))
```

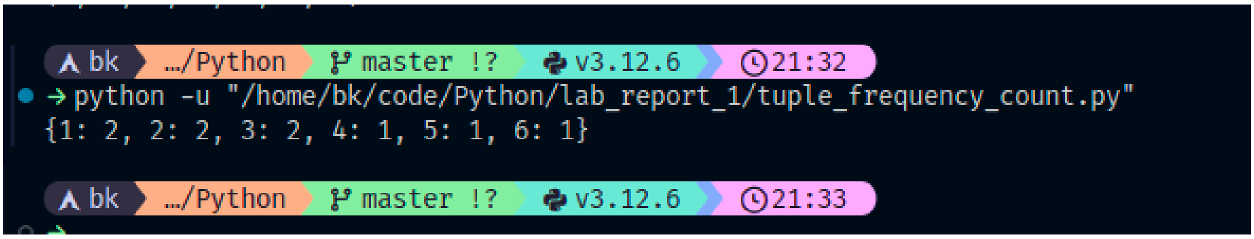


```
 bk  …/Python  master !?  v3.12.6  21:32
● → python -u "/home/bk/code/Python/lab_report_1/tuple_sort.py"
(0, 1, 2, 3, 4, 5, 7)

 bk  …/Python  master !?  v3.12.6  21:32
```

9. Tuple frequency

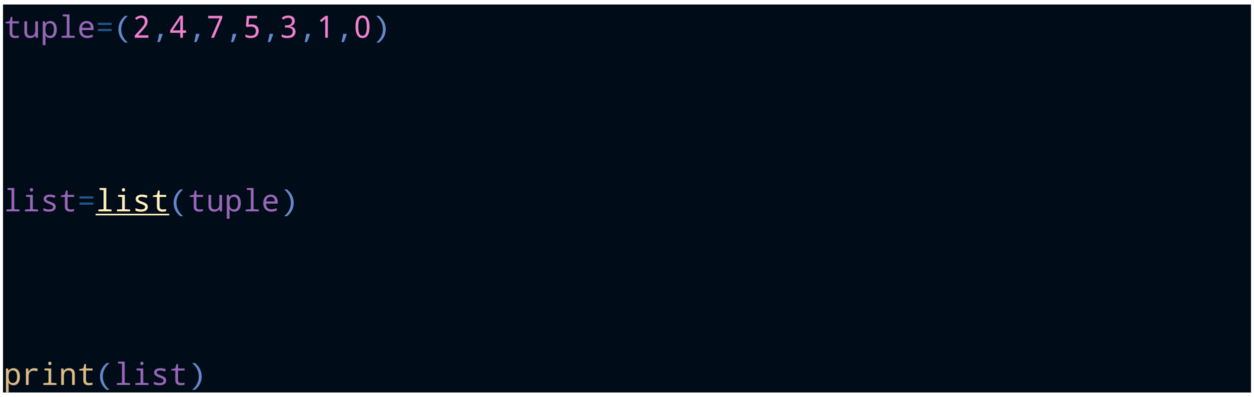


```
tuple=((1,2,3),(4,5,6),(1,2,3))
dict={}
for i in range(len(tuple)):
    for j in range(len(tuple[0])):
        if tuple[i][j] in dict:
            dict[tuple[i][j]]+=1
        else:
            dict[tuple[i][j]]=1


print(dict)
```

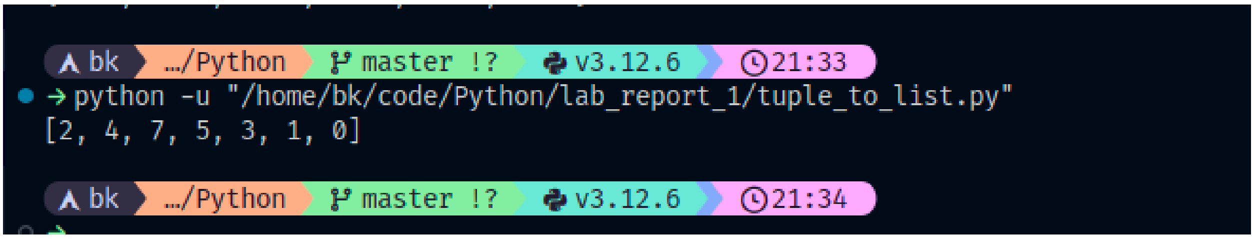


```
bk  …/Python  master !?  v3.12.6  21:32
→ python -u "/home/bk/code/Python/lab_report_1/tuple_frequency_count.py"
{1: 2, 2: 2, 3: 2, 4: 1, 5: 1, 6: 1}

bk  …/Python  master !?  v3.12.6  21:33
```

10. Tuple to list

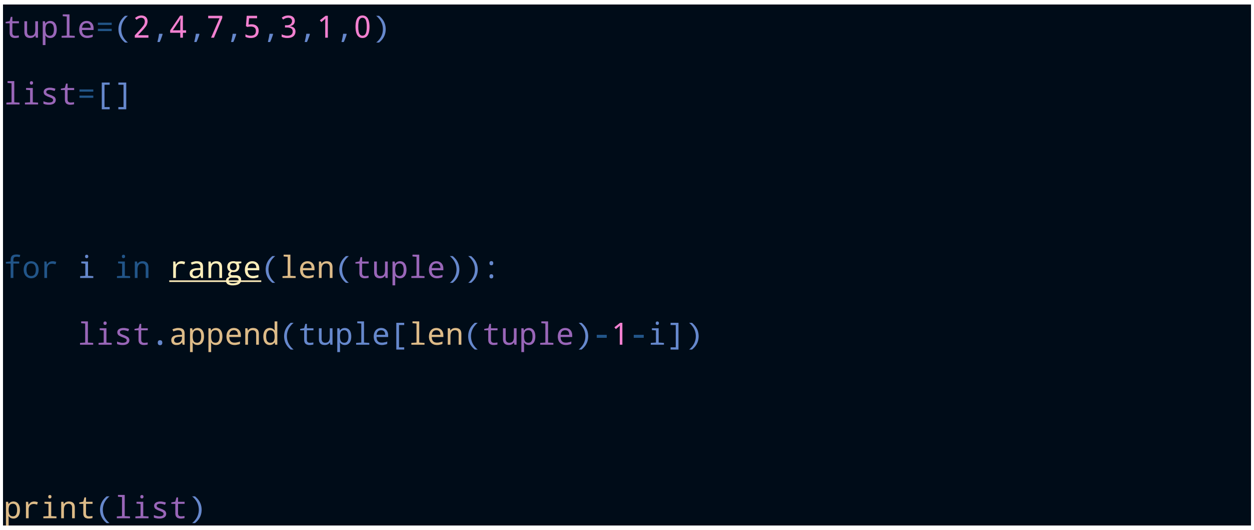


```python
tuple=(2,4,7,5,3,1,0)



list=list(tuple)



print(list)
```

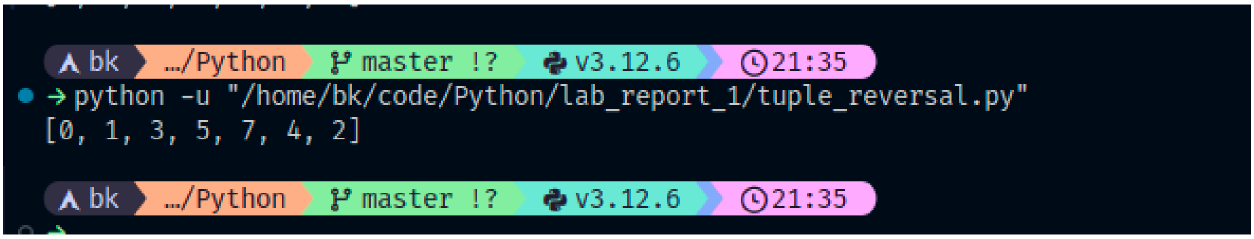


```
bk  …/Python  master !?  v3.12.6  21:33
→ python -u "/home/bk/code/Python/lab_report_1/tuple_to_list.py"
[2, 4, 7, 5, 3, 1, 0]

bk  …/Python  master !?  v3.12.6  21:34
```

11. Tuple reversal

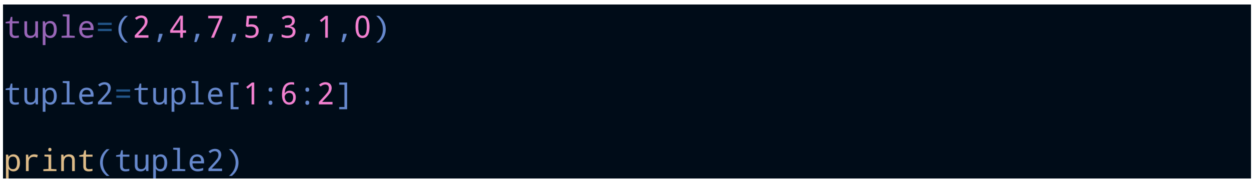


```python
tuple=(2,4,7,5,3,1,0)

list=[]



for i in range(len(tuple)):

    list.append(tuple[len(tuple)-1-i])



print(list)
```

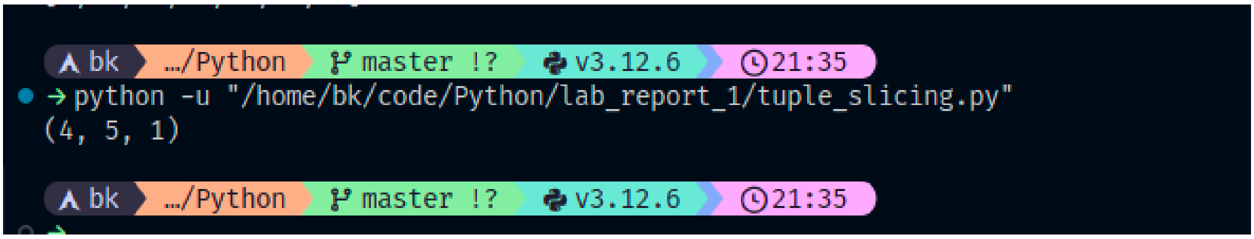


```
 bk  …/Python  master !?  v3.12.6  21:35
● → python -u "/home/bk/code/Python/lab_report_1/tuple_reversal.py"
 [0, 1, 3, 5, 7, 4, 2]

 bk  …/Python  master !?  v3.12.6  21:35
```

12. Tuple slicing

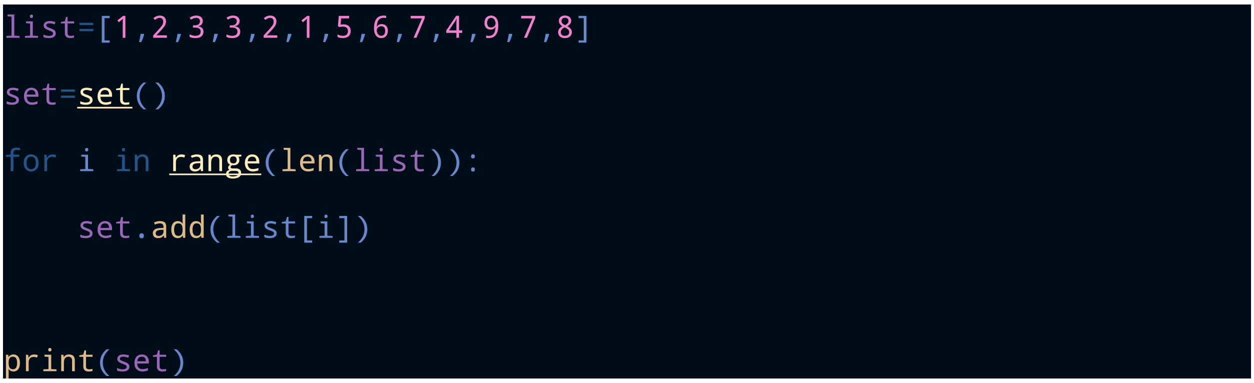


```
tuple=(2,4,7,5,3,1,0)

tuple2=tuple[1:6:2]

print(tuple2)
```

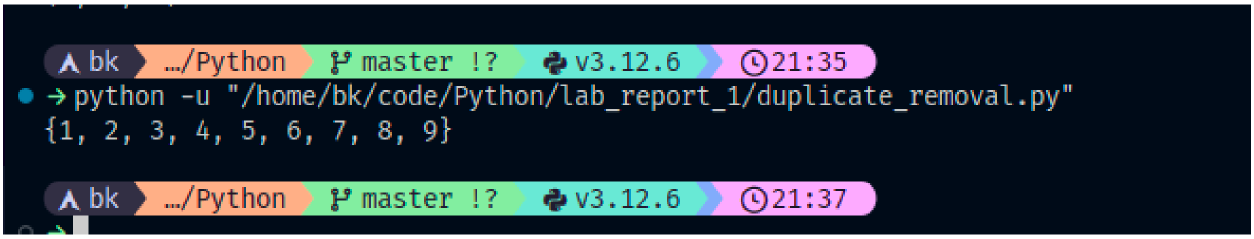


```
 bk  …/Python  master !?  v3.12.6  21:35
● → python -u "/home/bk/code/Python/lab_report_1/tuple_slicing.py"
 (4, 5, 1)

 bk  …/Python  master !?  v3.12.6  21:35
```

13. Duplicate removal

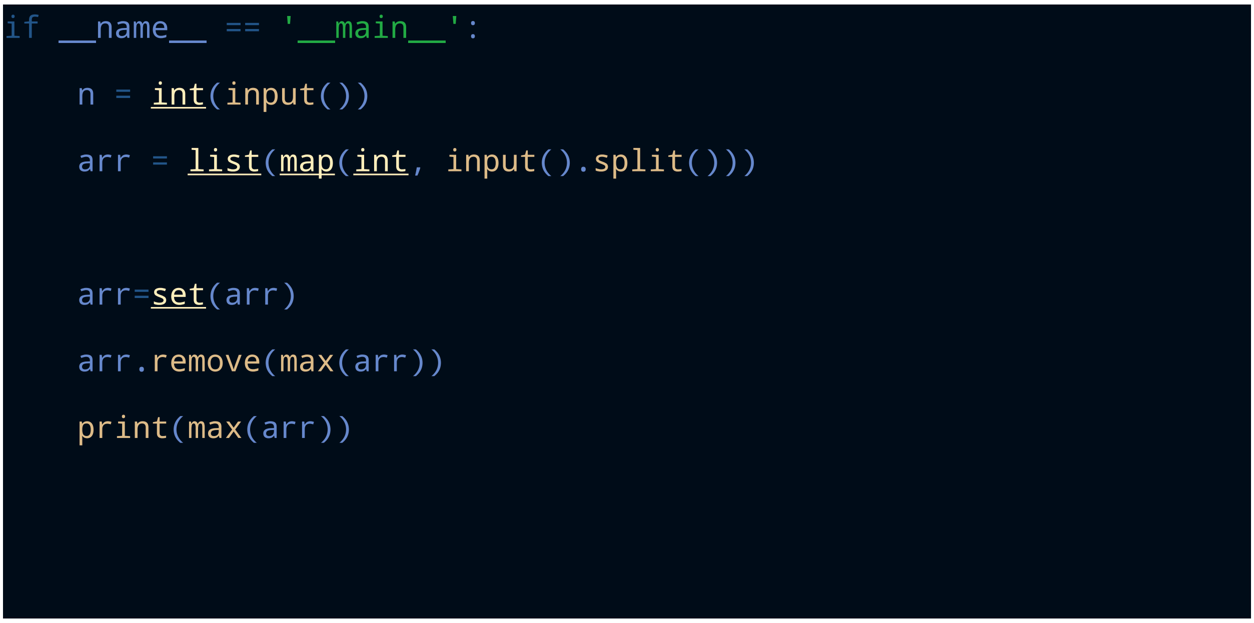


```
list=[1,2,3,3,2,1,5,6,7,4,9,7,8]

set=set()

for i in range(len(list)):

    set.add(list[i])


print(set)
```



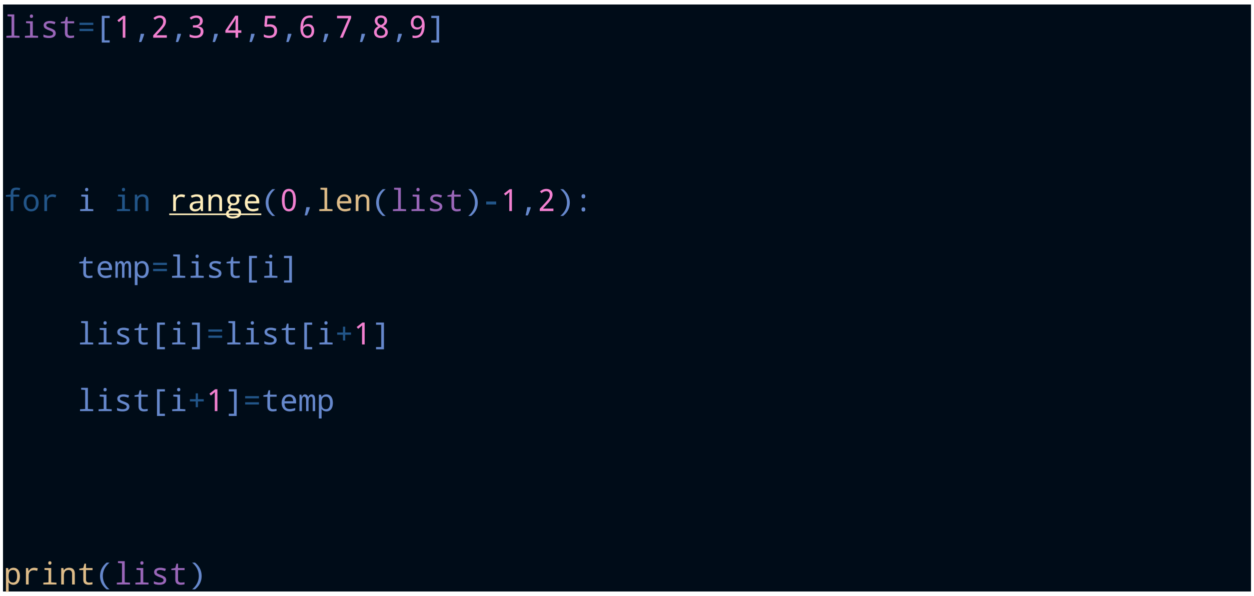
```
 bk  …/Python  master !?  v3.12.6  21:35
● → python -u "/home/bk/code/Python/lab_report_1/duplicate_removal.py"
 {1, 2, 3, 4, 5, 6, 7, 8, 9}

 bk  …/Python  master !?  v3.12.6  21:37
```

14. Second max number

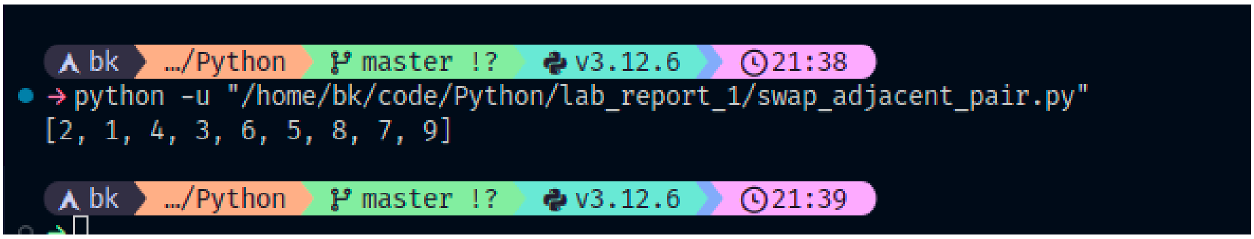


```python
if __name__ == '__main__':
    n = int(input())
    arr = list(map(int, input().split()))

    arr=set(arr)
    arr.remove(max(arr))
    print(max(arr))
```

15. Swap adjacent

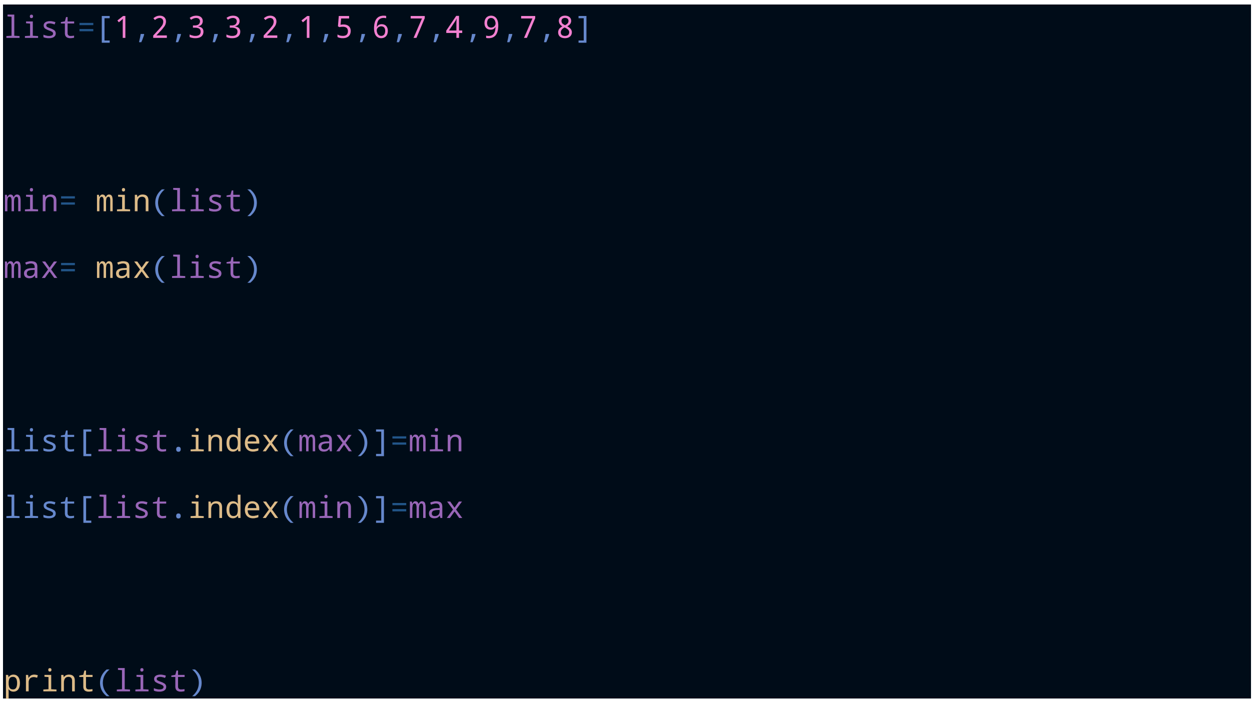


```python
list=[1,2,3,4,5,6,7,8,9]


for i in range(0,len(list)-1,2):
    temp=list[i]
    list[i]=list[i+1]
    list[i+1]=temp


print(list)
```

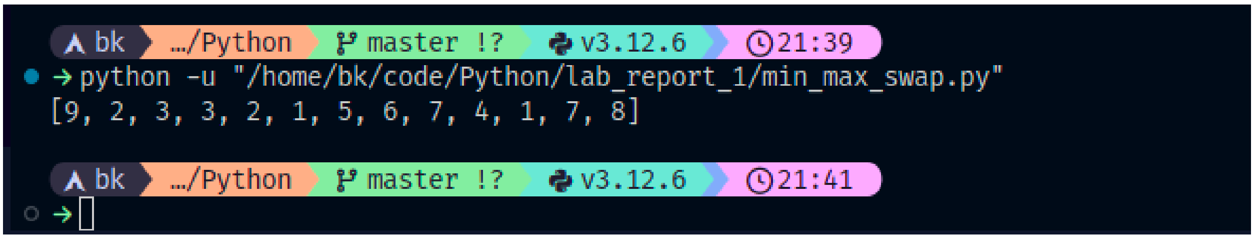


```
 bk  …/Python  master !?  v3.12.6  21:38
● → python -u "/home/bk/code/Python/lab_report_1/swap_adjacent_pair.py"
 [2, 1, 4, 3, 6, 5, 8, 7, 9]

 bk  …/Python  master !?  v3.12.6  21:39
```

16. Max min swap

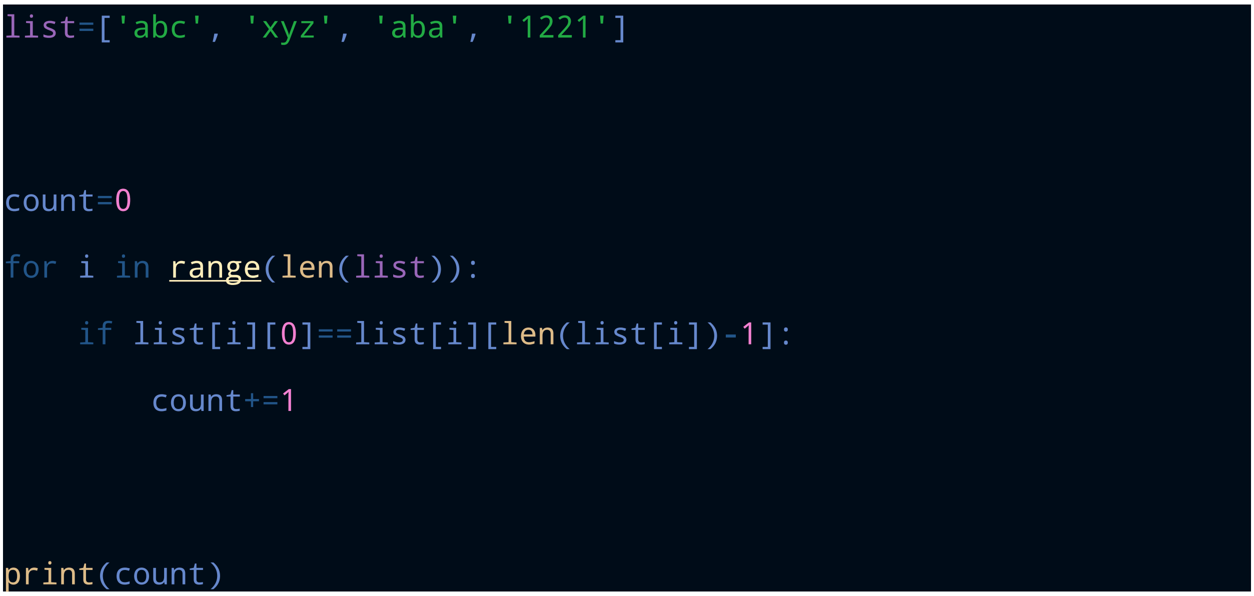


```python
list=[1,2,3,3,2,1,5,6,7,4,9,7,8]


min= min(list)

max= max(list)


list[list.index(max)]=min

list[list.index(min)]=max


print(list)
```

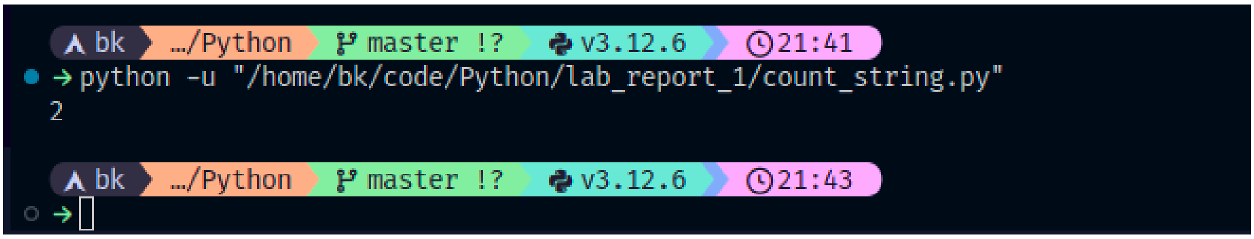


```
 bk  …/Python  master !?  v3.12.6  21:39
● → python -u "/home/bk/code/Python/lab_report_1/min_max_swap.py"
  [9, 2, 3, 3, 2, 1, 5, 6, 7, 4, 1, 7, 8]

 bk  …/Python  master !?  v3.12.6  21:41
○ →
```

17. Count string

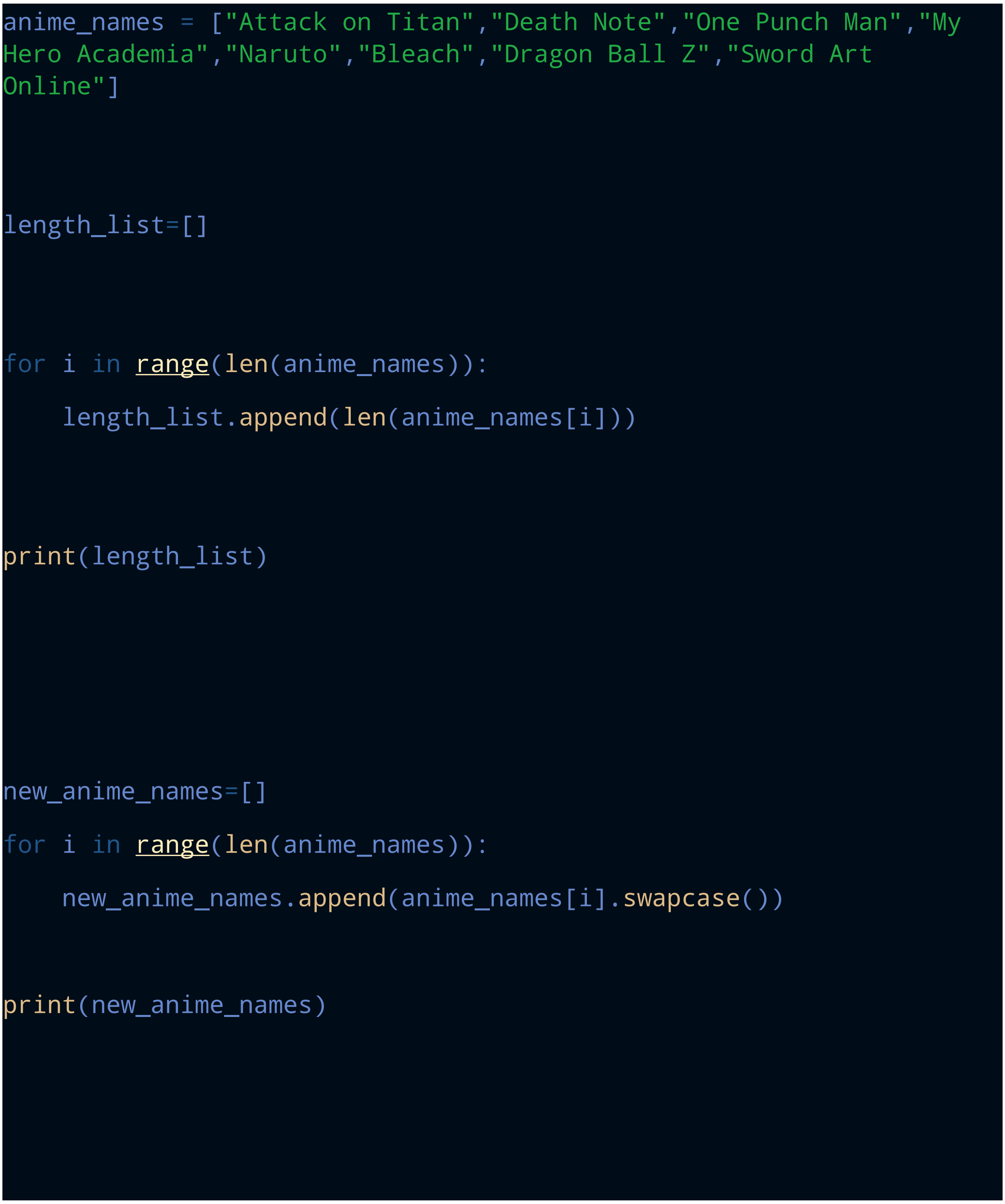


```python
list=['abc', 'xyz', 'aba', '1221']


count=0

for i in range(len(list)):

    if list[i][0]==list[i][len(list[i])-1]:

        count+=1


print(count)
```

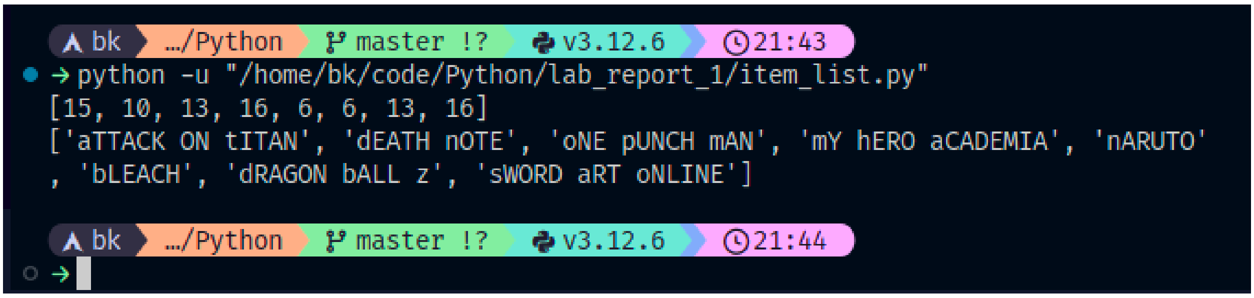


```
 bk  …/Python  master !?  v3.12.6  21:41
→ python -u "/home/bk/code/Python/lab_report_1/count_string.py"
2

 bk  …/Python  master !?  v3.12.6  21:43
→
```

18. Item List



```python
anime_names = ["Attack on Titan","Death Note","One Punch Man","My Hero Academia","Naruto","Bleach","Dragon Ball Z","Sword Art Online"]


length_list=[]


for i in range(len(anime_names)):
    length_list.append(len(anime_names[i]))


print(length_list)




new_anime_names=[]
for i in range(len(anime_names)):
    new_anime_names.append(anime_names[i].swapcase())

print(new_anime_names)
```



```
  bk  …/Python  master !?  v3.12.6  21:43
● → python -u "/home/bk/code/Python/lab_report_1/item_list.py"
  [15, 10, 13, 16, 6, 6, 13, 16]
  ['aTTACK ON tITAN', 'dEATH nOTE', 'oNE pUNCH mAN', 'mY hERO aCADEMIA', 'nARUTO'
  , 'bLEACH', 'dRAGON bALL z', 'sWORD aRT oNLINE']

  bk  …/Python  master !?  v3.12.6  21:44
○ →
```