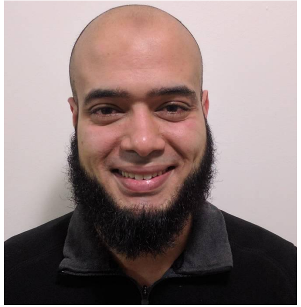
Python Programming

Dict Practice 1

Mostafa S. Ibrahim
Teaching, Training and Coaching since more than a decade!

Artificial Intelligence & Computer Vision Researcher
PhD from Simon Fraser University - Canada
Bachelor / Msc from Cairo University - Egypt
Ex-(Software Engineer / ICPC World Finalist)

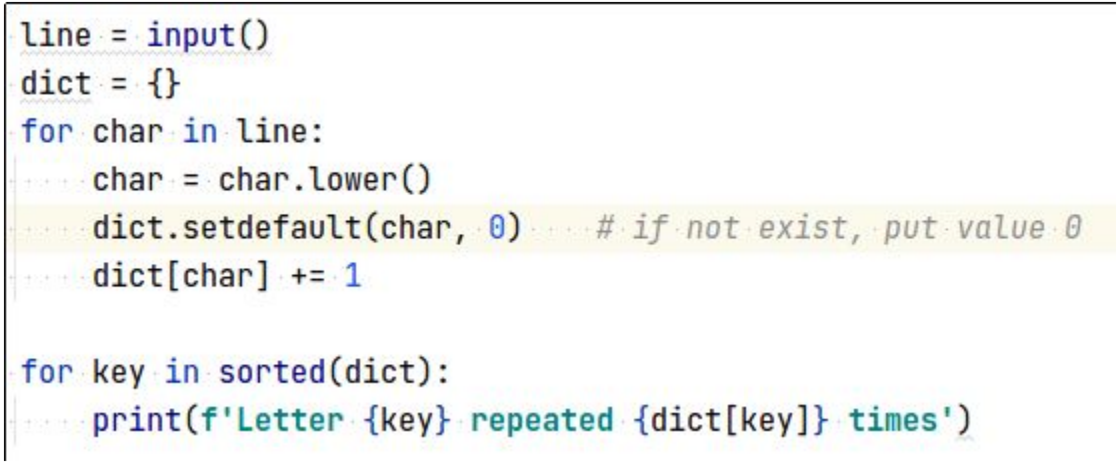


Practice 1: Letters Frequency

- Read a string of lower/upper letters.
- Convert to lowercase and then compute the frequency of letters.
    - Print from small to large.
- Input: bAAAaaazz
- Output (ordered based on key)
    - Letter a repeated 6 times
    - Letter b repeated 1 times
    - Letter z repeated 2 times
- Use dict in an efficient & simple way

Practice 1: Letters Frequency

- Convert to lower
- Get current letter value. It it doesn’t exist, set to 0
- Increment frequency
- Printed by sorted key



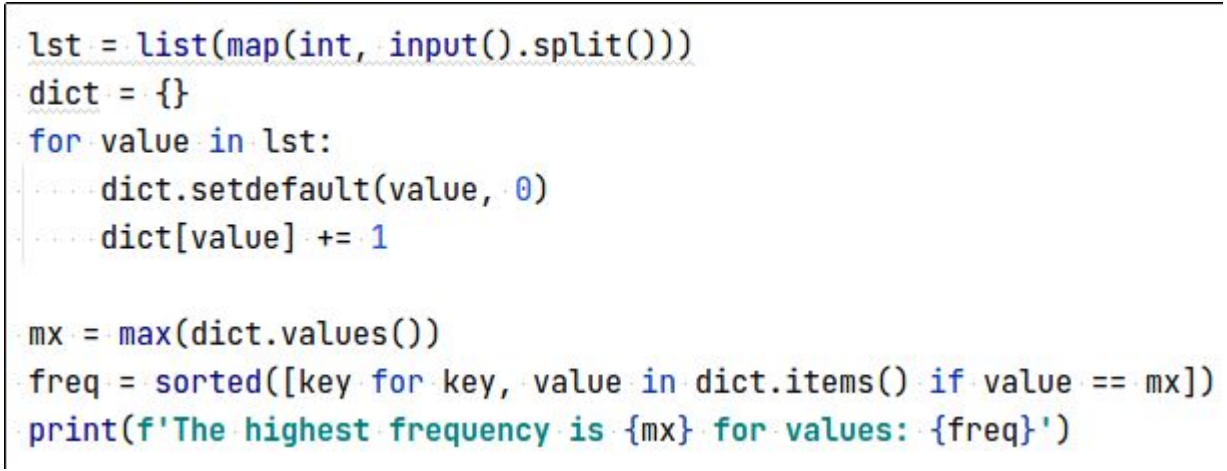
```python
line = input()
dict = {}
for char in line:
    char = char.lower()
    dict.setdefault(char, 0)    # if not exist, put value 0
    dict[char] += 1

for key in sorted(dict):
    print(f'Letter {key} repeated {dict[key]} times')
```

Practice 2: Find most frequent number

- Read a line of N integers. The values can be big and negative
- Find all the values that repeated the most number of times.
  - Print them from small to large
- Input: -123456 10 -123456 20 -30 -123456 20 25 20
- Output: The highest frequency is 3 for values: [-123456, 20]
- Don’t use nested loops

Practice 2: Find most frequent number

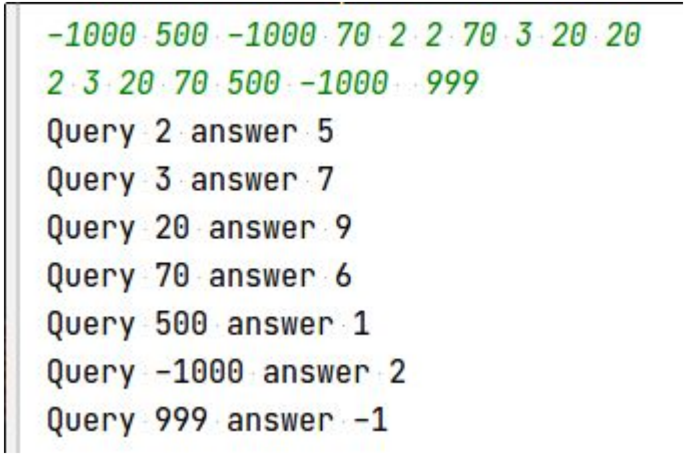
- We used list before, but the -ve value range was smaller (shift trick)
- Dict makes the problem trivial



```
lst = list(map(int, input().split()))
dict = {}
for value in lst:
    dict.setdefault(value, 0)
    dict[value] += 1

mx = max(dict.values())
freq = sorted([key for key, value in dict.items() if value == mx])
print(f'The highest frequency is {mx} for values: {freq}')
```

Practice 3: Search for a number

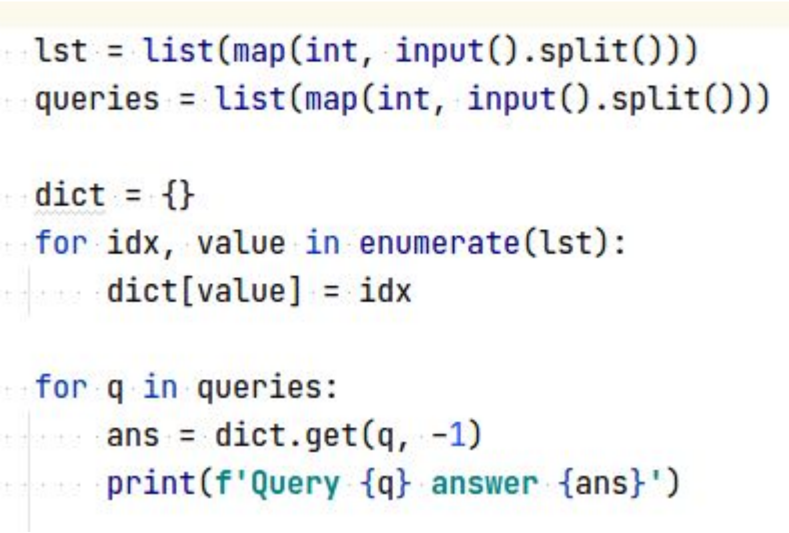
- Read a line of N integers, but the values can be big and negative
- Then read a line of Q integer
  - For each integer, print the index of the **last occurance** in the list or -1 if it doesn’t exist



```
-1000 500 -1000 70 2 2 70 3 20 20
2 3 20 70 500 -1000  999
Query 2 answer 5
Query 3 answer 7
Query 20 answer 9
Query 70 answer 6
Query 500 answer 1
Query -1000 answer 2
Query 999 answer -1
```

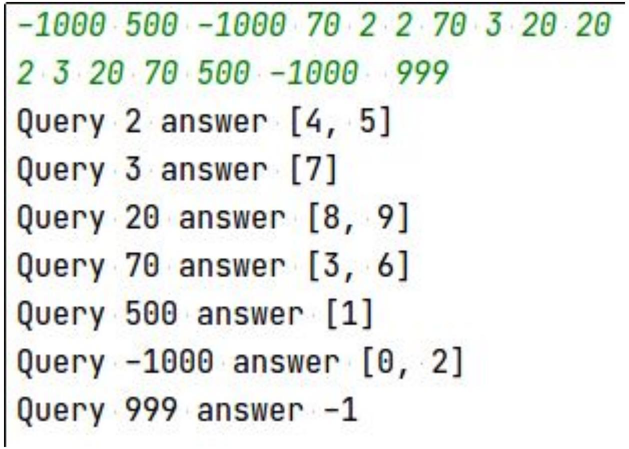
Practice 3: Search for a number

- By overriding the values, we easily know the last occurance

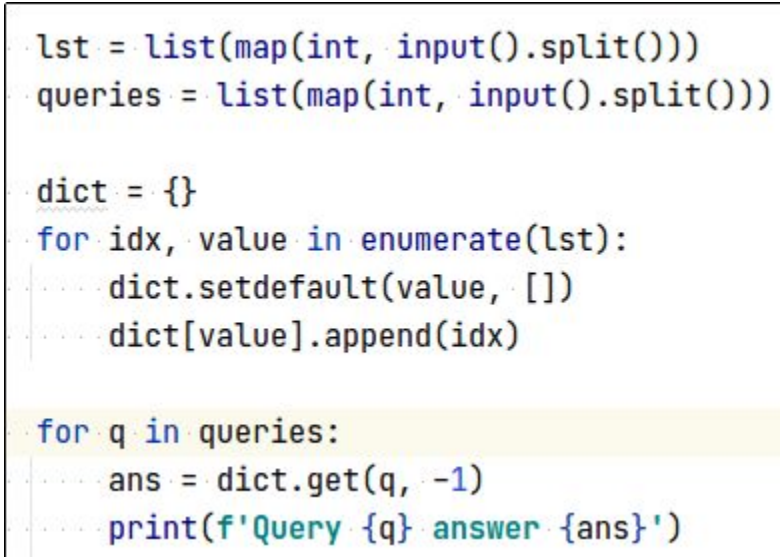


```python
lst = list(map(int, input().split()))
queries = list(map(int, input().split()))

dict = {}
for idx, value in enumerate(lst):
    dict[value] = idx

for q in queries:
    ans = dict.get(q, -1)
    print(f'Query {q} answer {ans}')
```

Practice 3: Search for a number: print all indices



```
-1000 500 -1000 70 2 2 70 3 20 20
2 3 20 70 500 -1000  999
Query 2 answer [4, 5]
Query 3 answer [7]
Query 20 answer [8, 9]
Query 70 answer [3, 6]
Query 500 answer [1]
Query -1000 answer [0, 2]
Query 999 answer -1
```



```python
lst = list(map(int, input().split()))
queries = list(map(int, input().split()))

dict = {}
for idx, value in enumerate(lst):
    dict.setdefault(value, [])
    dict[value].append(idx)

for q in queries:
    ans = dict.get(q, -1)
    print(f'Query {q} answer {ans}')
```

"Acquire knowledge and impart it to the people."

"Seek knowledge from the Cradle to the Grave."