

# Rock Collecting problem. ¶

[Go to site. \(https://prepinsta.com/tcs-codevita/python-code-for-counting-rock-samples-problem/\)](https://prepinsta.com/tcs-codevita/python-code-for-counting-rock-samples-problem/)

- Prepared by: [Sagun Shakya \(https://github.com/sagsshakya\)](https://github.com/sagsshakya)
  - MSc. Data Science
  - GITAM Institute of Science, Visakhapatnam.
  - Email: [sags.shakya@gmail.com](mailto:sags.shakya@gmail.com) (<mailto:sags.shakya@gmail.com>)
- 
- Question – : Juan Marquinho is a geologist and he needs to count rock samples in order to send it to a chemical laboratory. He has a problem: The laboratory only accepts rock samples by a range of its size in ppm (parts per million).
  - Juan Marquinho receives the rock samples one by one and he classifies the rock samples according to the range of the laboratory. This process is very hard because the number of rock samples may be in millions.
  - Juan Marquinho needs your help, your task is to develop a program to get the number of rocks in each of the ranges accepted by the laboratory.
  - Input Format: An positive integer S (the number of rock samples) separated by a blank space, and a positive integer R (the number of ranges of the laboratory); A list of the sizes of S samples (in ppm), as positive integers separated by space R lines where the ith line containing two positive integers, space separated, indicating the minimum size and maximum size respectively of the ith range.
  - Output Format: R lines where the ith line contains a single non-negative integer indicating the number of the samples which lie in the ith range.

## Getting inputs as per the question.

```
In [23]: myinput = input('Enter the number of samples and the number of ranges separated by a space: ')
Enter the number of samples and the number of ranges separated by a space: 20 3
```

```
In [24]: S, R = myinput.split()
S, R= int(S), int(R)
```

```
In [26]: S,R
```

```
Out[26]: (20, 3)
```

```
In [25]: range_list = []
         for ii in range(R):
             temp = input('Enter range separated by a space: ')
             l1, l2 = temp.split()
             l1, l2 = int(l1), int(l2)
             range_list.append((l1, l2))
```

Enter range separated by a space: 1 100  
Enter range separated by a space: 50 600  
Enter range separated by a space: 1 1000

```
In [27]: range_list
```

```
Out[27]: [(1, 100), (50, 600), (1, 1000)]
```

```
In [28]: samples = input('Enter the sample values separated by a space: ')
```

Enter the sample values separated by a space: 921 107 270 631 926 543 589 520 5  
95 93 873 424 759 537 458 614 725 842 575 195

```
In [29]: samples_list = samples.split()
         samples_list = [int(sample) for sample in samples_list]
         sorted_list = sorted(samples_list)
         sorted_list
```

```
Out[29]: [93,
          107,
          195,
          270,
          424,
          458,
          520,
          537,
          543,
          575,
          589,
          595,
          614,
          631,
          725,
          759,
          842,
          873,
          921,
          926]
```

**Keepig track of the counts.**

```
In [30]: counter = dict()
         for ii in range(R):
             counter['sum' + str(ii+1)] = 0
```

**Keeping track of the items.**

```
In [31]: identifier = dict()
        for ii in range(R):
            identifier['ranger' + str(ii+1)] = []
```

**Main calculation part.**

```
In [32]: for sample in sorted_list:
        for ii in range(len(range_list)):
            if sample in range(*range_list[ii]):
                counter['sum' + str(ii+1)] += 1
                identifier['ranger' + str(ii+1)].append(sample)
```

**Outputs.**

```
In [34]: counter
```

```
Out[34]: {'sum1': 1, 'sum2': 12, 'sum3': 20}
```

```
In [36]: print(identifier)
```

```
{'ranger1': [93], 'ranger2': [93, 107, 195, 270, 424, 458, 520, 537, 543, 575, 589, 595], 'ranger3': [93, 107, 195, 270, 424, 458, 520, 537, 543, 575, 589, 595, 614, 631, 725, 759, 842, 873, 921, 926]}
```

```
In [38]: print('Your answer is: \n')
        for value in counter.values():
            print(value)
```

Your answer is:

```
1
12
20
```

**The End.**