

In [6]:

```
1 #1
2
3 class First:
4     def __init__(self,n):
5         self.n=n
6     def gene(self):
7         for i in range(1,self.n+1):
8             if i%7==0:
9                 yield i
10
11
12 n=int(input())
13
14 g=First(n)
15 g.gene()
16
17 for i in g.gene():
18     print(i)
```

```
70
7
14
21
28
35
42
49
56
63
70
```

In [8]:

```
1 # 2
2
3 n=tuple(input().split(" "))
4
5 for i in set(n):
6     print(str(i)+" ":" "+str(n.count(i)))
```

New to Python or choosing between Python 2 and Python 3? Read Python 2 or Python 3

New:1

or:2

between:1

3?:1

choosing:1

Python:5

2:2

to:1

and:1

Read:1

3:1

In [11]:

```
1 class Person:
2
3     def getGender(self):
4         pass
5 class Male(Person):
6     def getgender(self):
7         print("Male")
8 class Female(Person):
9     def getgender(Person):
10        print("Female")
11
12 male=Male()
13 male.getgender()
14 female=Female()
15 female.getgender()
```

Male

Female

In [12]:

```
1 # 4
2
3 subjects = ["I", "You"]
4 verbs = ["Play", "Love"]
5 objects = ["Hockey", "Football"]
6
7 sentences = []
8
9 for subject in subjects:
10     for verb in verbs:
11         for obj in objects:
12             sentence = subject + " " + verb + " " + obj + "."
13             sentences.append(sentence)
14
15
16 for sentence in sentences:
17     print(sentence)
18
```

I Play Hockey.
I Play Football.
I Love Hockey.
I Love Football.
You Play Hockey.
You Play Football.
You Love Hockey.
You Love Football.

In [14]:

```
1 # 5
2 s="hello world!hello world!hello world!hello world"
3 k=s.encode()
4 print(k)
5 print(k.decode())
```

b'hello world!hello world!hello world!hello world'
hello world!hello world!hello world!hello world

In [18]:

```
1 # 5
2
3 import zlib
4 s="hello world!hello world!hello world!hello world"
5 k=zlib.compress(s.encode())
6 print(k)
7 z=zlib.decompress(k).decode()
8 print(z)
```

b'\x9c\xcbH\xcd\xc9\xc9W(\xcf/\xcaIQ\xcc \xcc\x06\x00\xabf\x11\xd4'
hello world!hello world!hello world!hello world

In [19]:

```
1 # 6
2
3 def binary_search(arr, target):
4     left = 0
5     right = len(arr) - 1
6
7     while left <= right:
8         mid = (left + right) // 2
9
10        if arr[mid] == target:
11            return mid
12        elif arr[mid] < target:
13            left = mid + 1
14        else:
15            right = mid - 1
16
17    return -1
18
19
20
21 sorted_list = [1, 3, 5, 7, 9, 11, 13]
22 target = 7
23
24 index = binary_search(sorted_list, target)
25 if index != -1:
26     print(f"Element {target} found at index {index}")
27 else:
28     print(f"Element {target} not found in the list")
29
```

Element 7 found at index 3

