





```
86  ::***::
87
88  All tests passed
89
90  ✨🌟💎🦄💎💎🌟🌟🌟💎🦄💎💎🌟🌟
91
92  Points 30/30
```



 **SuperDoggez** qwerty ✕

🔍 2 contributors  

10 lines (7 sloc) | 198 Bytes


```
1  import numpy as np
2
3
4  def Problem1(n):
5      onesk = np.ones((n,n))
6      indx = [(x,y) for x in range(n-1) for y in range(x+1,n)]
7      xyindx = tuple(zip(*indx))
8      onesk[xyindx] = 0
9      return onesk
```



 SuperDoggez Final ✓

🔍 2 contributors  

16 lines (13 sloc) | 468 Bytes

```
1 def Reduce_this_to_NestedListComprehension(N):
2     primes = []
3     for num in range(1,N):
4         list_of_divisables = []
5         for d in range(2,num):
6             if num%d == 0: #divisible by some number
7                 list_of_divisables.append(d)
8             if not list_of_divisables:
9                 primes.append(num)
10    return primes
11
12
13 def Problem2(N):
14     primes = [num for num in range (1,N) if not [d for d in range(2,num) if num%d==0]]
15     return primes
```

 SuperDoggez Final ✓

🔍 2 contributors  

5 lines (4 sloc) | 229 Bytes

```
1 def Problem3(N):
2     primes = [num for num in range (1,N) if not [d for d in range(2,num) if num%d==0]]
3     odd_dict_of_primes = {i:j for (i,j) in dict(zip(range(len(primes)),primes)).items() if i%2!=0}
4     return odd_dict_of_primes
```