

Generate and Test Paradigm: [3,100]

3 is prime	22 is NOT prime	41 is prime	60 is NOT prime
4 is NOT prime	23 is prime	42 is NOT prime	61 is prime
5 is prime	24 is NOT prime	43 is prime	62 is NOT prime
6 is NOT prime	25 is NOT prime	44 is NOT prime	63 is NOT prime
7 is prime	26 is NOT prime	45 is NOT prime	64 is NOT prime
8 is NOT prime	27 is NOT prime	46 is NOT prime	65 is NOT prime
9 is NOT prime	28 is NOT prime	47 is prime	66 is NOT prime
10 is NOT prime	29 is prime	48 is NOT prime	67 is prime
11 is prime	30 is NOT prime	49 is NOT prime	68 is NOT prime
12 is NOT prime	31 is prime	50 is NOT prime	69 is NOT prime
13 is prime	32 is NOT prime	51 is NOT prime	70 is NOT prime
14 is NOT prime	33 is NOT prime	52 is NOT prime	71 is prime
15 is NOT prime	34 is NOT prime	53 is prime	72 is NOT prime
16 is NOT prime	35 is NOT prime	54 is NOT prime	73 is prime
17 is prime	36 is NOT prime	55 is NOT prime	74 is NOT prime
18 is NOT prime	37 is prime	56 is NOT prime	75 is NOT prime
19 is prime	38 is NOT prime	57 is NOT prime	76 is NOT prime
20 is NOT prime	39 is NOT prime	58 is NOT prime	77 is NOT prime
21 is NOT prime	40 is NOT prime	59 is prime	78 is NOT prime
22 is NOT prime	41 is prime	60 is NOT prime	79 is prime

80 is NOT prime
81 is NOT prime
82 is NOT prime
83 is prime
84 is NOT prime
85 is NOT prime
86 is NOT prime
87 is NOT prime
88 is NOT prime
89 is prime
90 is NOT prime
91 is NOT prime
92 is NOT prime
93 is NOT prime
94 is NOT prime
95 is NOT prime
96 is NOT prime
97 is prime
98 is NOT prime
99 is NOT prime
100 is NOT prime

Prime numbers in range [3,100] is: [3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89, 97]

Composite numbers in range [3,100] is: [4, 6, 8, 9, 10, 12, 14, 15, 16, 18, 20, 21, 22, 24, 25, 26, 27, 28, 30, 32, 33, 34, 35, 36, 38, 39, 40, 42, 44, 45, 46, 48, 49, 50, 51, 52, 54, 55, 56, 57, 58, 60, 62, 63, 64, 65, 66, 68, 69, 70, 72, 74, 75, 76, 77, 78, 80, 81, 82, 84, 85, 86, 87, 88, 90, 91, 92, 93, 94, 95, 96, 98, 99, 100]

Generate and Test Paradigm: User Input Intervals

Good input intervals:

```
Please enter starting prime interval: 2
Please enter ending prime interval: 10
2 is prime
3 is prime
4 is NOT prime
5 is prime
6 is NOT prime
7 is prime
8 is NOT prime
9 is NOT prime
10 is NOT prime
```

Prime numbers in range [2 , 10] is:

```
[2, 3, 5, 7]
```

Composite numbers in range [2 , 10] is:

```
[4, 6, 8, 9, 10]
```

Bad input intervals:

Please enter starting prime interval: 1

```
-----
ValueError                                Traceback (most recent call last)
/var/folders/g6/m9mjx8gn46xg9qvcz4921f_c0000gn/T/ipykernel_2108/1754207646.py in <module>
    28     start_interval_input = int(input("Please enter starting prime interval: "))
    29     if start_interval_input <= 1:
--> 30         raise ValueError("Value cannot be <= 1")
    31     start_interval_input = None
    32
```

ValueError: Value cannot be <= 1

Please enter starting prime interval: 3
Please enter ending prime interval: -1

```
-----
ValueError                                Traceback (most recent call last)
/var/folders/g6/m9mjx8gn46xg9qvcz4921f_c0000gn/T/ipykernel_2108/2131392177.py in <module>
    36     end_interval_input = int(input("Please enter ending prime interval: "))
    37     if end_interval_input < start_interval_input:
--> 38         raise ValueError("End value must be higher than start value")
    39     if end_interval_input <= 1:
    40         raise ValueError("Value cannot be <= 1")
```

ValueError: End value must be higher than start value