

```
def isPrime(n):
    half=n//2
    if n < 2:
        return False
    for i in range(2, half+1):
        if n % i == 0:
            return False
    return True
```

Approach-

Any number can be considered as prime number when mod of the number with any number starting from 2 until half of the number + 1 is 0. All the case statements are written to depict this concept.

```
print(isPrime(6))
```

False

```
##Using loop until sqrt of the number
```

```
import math
```

```
def isPrime2(n):
    sqrt=round(math.sqrt(n))

    # print(sqrt)

    if n < 2:
        return False
    for i in range(2, sqrt+1):
        if n % i == 0:
            return False
    return True
```

Approach-

Any number can be considered as prime number when mod of the number with any number starting from 2 until square root of the number + 1 is 0. All the case statements are written to depict this concept.

```
print(isPrime2(50))
```

False

```
## Find 9991th to 10000th prime numbers and then write to prime.txt file
```

```
from google.colab import drive
drive.mount('/content/drive')
```

```
f = open("/content/drive/MyDrive/Colab Notebooks/prime.txt", "w")
```

```
counter=0
for i in range(1, 1000000):
    if isPrime2(i):
        counter+=1

    if counter >= 9991 :
        if counter <= 10000 :
            print(str(i) + ' is the ' + str(counter) + 'th prime number')
            f.write(str(i) + ' is the ' + str(counter) + 'th prime number\n')
            continue ## Until the prime number counter is between 9991th and 10000th, then the loop will continue.
        else:
            break ## if the prime number counter goes above 10000th, then the loop will break.
f.close()
```

Mounted at /content/drive

```
104677 is the 9991th prime number
104681 is the 9992th prime number
104683 is the 9993th prime number
104693 is the 9994th prime number
104701 is the 9995th prime number
104707 is the 9996th prime number
104711 is the 9997th prime number
```

```
104717 is the 9998th prime number  
104723 is the 9999th prime number  
104729 is the 10000th prime number
```

Approach-

- a. Created a loop which runs between 1 to 1000000.
- b. Used a counter initialized to 0. The counter will increase over inside the loop.
- c. Called isPrime2 method and passed over numbers 1 to 1000000 through it. If the number is prime, then I am checking if the prime number is the 9991th till 10000th prime number using couple of if cases.