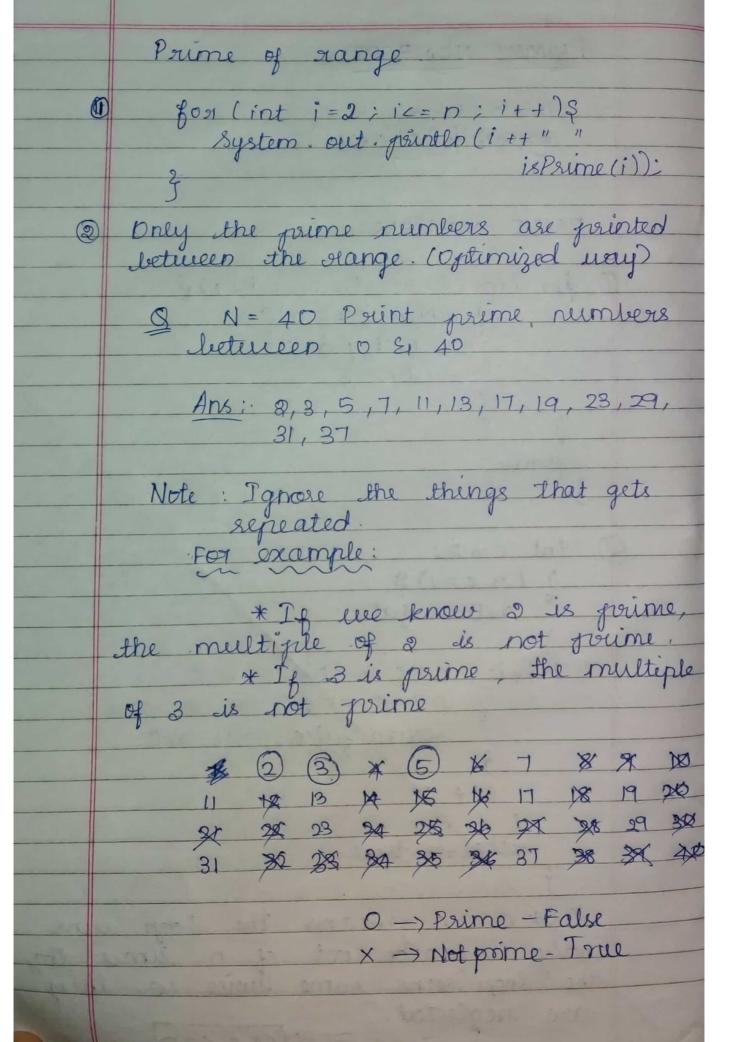
```
PRIME NUMBERS :
 O for lint isa; ic N; i++) §
         ij (N) i == 0)$
              Not prime:
              break;
    prime;
   int (= 2;
    ig (n <= 1) $
       return false:
    while (c*c <=n) }
         g (01.C == 0)
          setuen false:
ett;
    neturn true;
  11 C* C <= n means the loop suns
  the loop seems same twice so, they
  vare reglected.
           C <= \( \operatorn = \) (* C <= \( \operatorn \)
```



\* When you eliminate all multiples of 8, then only little bit multiples of 3 will get eliminated.

for example:

⇒ Via &, when we eliminate 18, we don't need to eliminate 18 again when multiples of 3 getting eliminated.

is itself not a prime, i.e., the factor of 6 has appeared before, no need to scheck for it.

grime since one of its factor has

appeared before.

loop only sort (n) times that will ignore the seperated case.

This is known as diene of Exatosthenes Algorithm. Which is an ancient algorithm that is used to find all the primes less than given number N.

## TIME COMPLEXITY:

How many numbers exist in a startsible by a. Hers say n=40 nots divisible by a at -13.

By 5 = 40 = 8 . A.

ie.,  $\frac{N}{2} + \frac{N}{3} + \frac{N}{5} + \frac{N}{7} + \cdots + \frac{N}{p}$ Where p = Highest prime that isless than N= N ( \frac{1}{2} + \frac{1}{3} + \frac{1}{5} + \frac{1}{7} + \frac{1}{p}) Solving by HP

Harmonic Progression for prime
is log(log N) =: N (log(log N)) .. Time Complexity = O(N(log(log(N)))