Todays Content

1. Prime
2. Prime Seive
3. Count of Factors

Prime: N is prime, if it has 2 factors

Ex: 11, 3, 5 ..

Q: Given N, check if N is prime or Not?

Ex: N=10 ✗ return false

N=11 ✓ return true

#Idea Calculate factors of N & compare == 2.

Idea1: Iterate from 1.. N & calculate factors == 2

TC: O(N) SC: O(1)

Idea2: Factors come in pairs

i is factor N/i is also factor

N=36

i	<=	N/i	
1		36	+2
2		18	+2
3		12	+2
4		9	+2
6		6	+1
9		4	
12		3	
18		2	
36		1	

$i=1;\ i <= N/i \Rightarrow i^2 <= N \Rightarrow i <= \sqrt{N}$

```
bool isPrime(int N){ TC: O(√N)
    int c=0;
    for(int i=1; i*i <= N; i++){
        if(N%i==0){ # i & N/i
            if(i==N/i){ c++;}
            else{ c+=2;}
        }
    }
    return c==2;
}
```

Idea3: N is prime 2 factors 1 & itself

[~~1~~ 2 .. √N .... ~~N~~]

$\boxed{\text{If N is prime: It will have no prime factors from } 2..\sqrt{N}}$

2Q Given N, print all primes from 1.. N

N = 10 : Output = 2 3 5 7

N = 15 : Output = 2 3 5 7 11 13

Ideas: Iterate from 1.. N;
check if a number is prime or not & print accordingly.

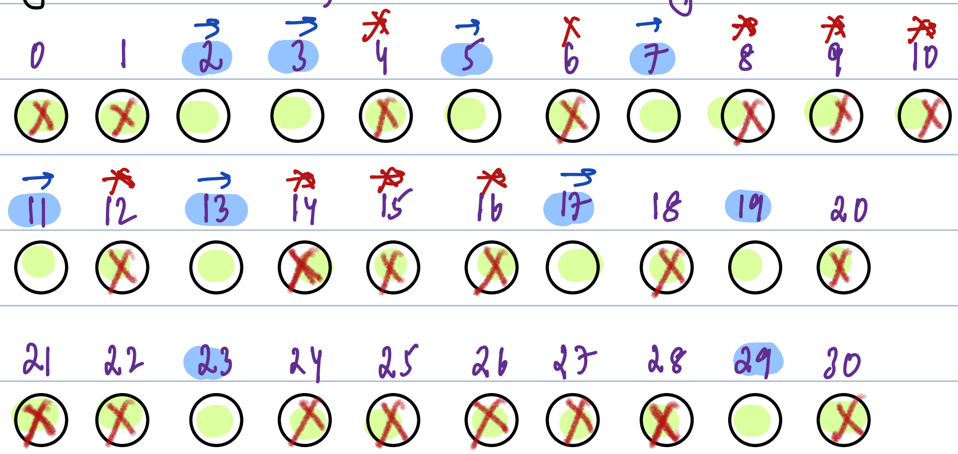
```
void allPrimes(int N){ TC: O(N√N) SC: O(1)
    for(int i=1; i<=N; i++){
        if( isPrime(i)){
            print(i);
        }
    }
}

bool isPrime(int N){ TC: O(√N)
    int c=0;
    for(int i=1; i*i<=N; i++){
        if(N%i==0){ # i & N/i
            if(i==N/i){ c++;}
            else{ c+=2;}
        }
    }
    return c==2;
}
```

Idea2: Seive Erthosthenes / Prime Seive

1. Assume all numbers from 1..N are prime.
2. If a number i is prime → It's a prime
   It's multiples of i = i 2i 3i 4i 5i... are not prime

Ex: N=30

Dry Run: Ex: N=30, create N+1 array



Pseudo Code:

1. Assume all elements from 1..N are primes
2. i = 2 .. N:
   - if i is prime:
     - Iterate on multiple of i & strike them
3. i = 2..N : Print primes

```
void allPrimes(int N){    TC: O(N log log_2 N)   SC: O(N)
    vector<bool> p(N+1, true);
    p[0] = p[1] = false;

    int c=0;
    for(int i=2; i<=N; i++){
        if(p[i]){  # i is prime: Multiples of i are
                   not prime, strike them
            c++;
            for(int j=2; j*i<=N; j++){
                p[j*i] = false;
            }
        }
    }
    return c;
}
```

j	j*i
2	2i
3	3i
4	4i

Iterations

i	Iterations
2	Mul of 2 till N; N/2
3	Mul of 3 till N; N/3
4	*
5	Mul of 5 till N; N/5
6	*
7	Mul of 7 till N; N/7
⋮	
p	Mul of p till N; N/p

↳ greater prime <= N

Total iterations = $O(N + N \log \log_2 N) = O(N \log \log_2 N)$

Outer loop = N

Inner loop : $N/2 + N/3 + N/5 + N/7 + \dots N/p$

Inner loop : $N/2 + N/3 + N/5 + N/7 + \cdots N/p$

$= N[1/2 + 1/3 + 1/5 + 1/7 + \cdots 1/p]$

Sum of reciprocals of prime till N : $\log_2^{\log_2^N}$, $\log_2 \log^N$

$= N * \log \log_2^N$

2Q Given N for all numbers from 1... N, print it's count of factors

Ex:

N = 10 : 1 2 3 4 5 6 7 8 9 10

Output 1 2 2 3 2 4 2 4 3 4

Idea: Iterate from 1.. N:

Calculate no: of factors & print it.

↳ Can be done in $\sqrt{N}$ time

TC: $O(N\sqrt{N})$ SC: $O(1)$ TODO

```
int factors(int N){

}
```

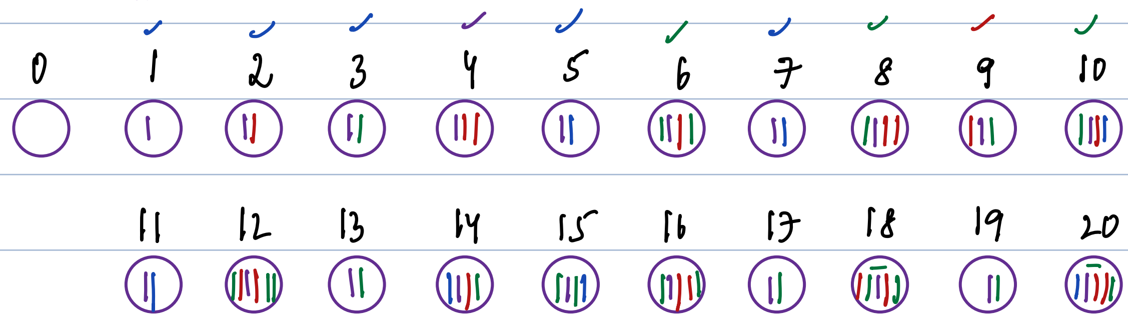
```
void Allfactor(int N){    TC:O(  )  SC:O( )

}
```

#Idea2: Given N, create cnt[N+1] & Initialize = 1.

i = 2 .. N;

Iterate on multiples of i = {i, 2i, 3i ..} & Inc count;

N = 20



void factors(int N){ TC: $O(N \log_2 N)$ SC: O(N)

```
void factors(int N){
    vector<int> fc(N+1, 1);
    for(int i=2; i<=N; i++){
        for(int j=1; j*i<=N; j++){
            fc[i*j]++;
        }
    }
    for(int i=1; i<=N; i++){
        print(fc[i]);
    }
}
```

Iterations

i	Iterations
2	Mul of 2 b/w N = $N/2$
3	Mul of 3 b/w N = $N/3$
4	Mul of 4 b/w N = $N/4$
⋮	
N	Mul of N b/w N = N/N

Total Iteration = $N + N\log_2 N$

Outer loop = N

Inner loop = $N/2 + N/3 + N/4 + \dots N/N$

$= N\underbrace{[1/2 + 1/3 + 1/4 + \dots 1/N]}$

Sum of reciprocals of natural nums = $\log_{(1)} N$

$= N\log_2 N$

Segmented Seive:

Given [a, b] find no: of primes in range [a..b]

Constraints:

$1 <= a <= b <= 10^9$

$1 <= b - a <= 10^5$

Idea1: Calculate primes from [1.. b]:

Iterate from a..b:

if number is prime: cnt

Issue: In above we will create bool p[b+1];

$b <= 10^9 \longrightarrow p[10^9]$ 1GB memory

$10^3 B = 1KB$

$10^6 B = 1MB$

$10^9 B = 1GB$

Idea2: TODO Later

Ideas:

Ex: $a = 100$ $b = 130$

6

Observations

ele index

Generalize $\{a..b\}$

i^{th} multiple of i: [ ]

i^{th} mul of 2 from [100 130]

i^{th} mul of 3 from [100 130]

i^{th} mul of 5 from [100 130]

i^{th} mul of 7 from [100 130]

i^{th} mul of 11 from [100 130]

i^{th} mul of i from [a .. b]

Generalize for i:

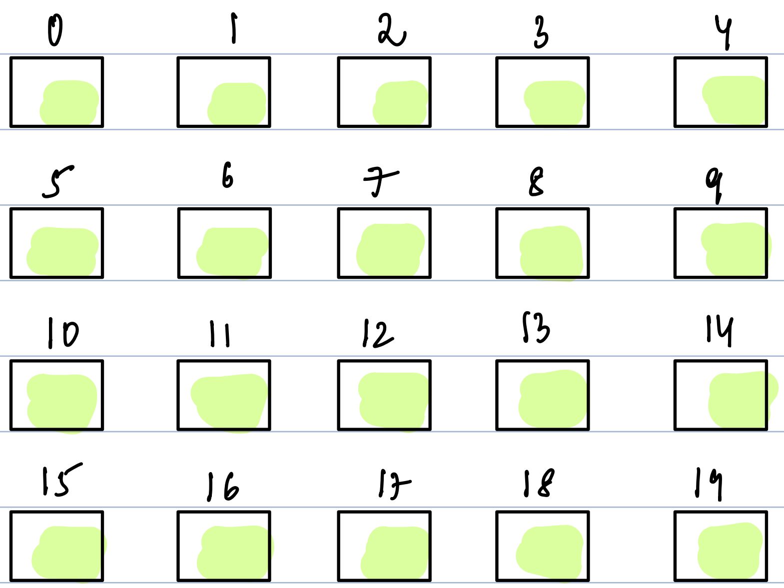
```
int segmentSeive(int a, int b){

}
```

Issues in above code:

Ex: $a = 1$ $b = 20$

Observations

ele Index



Issue:

i^{th} mul of 2 from $[1 \quad 20]$ ($a = 1$, $b = 20$)

i^{th} mul of 3 from $[1 \quad 20]$