Will Start at 9:05pm

Today's Agenda :-

1) Intro to Prime Numbers

2) Get all primes from 1 to N (Sieve of Eratosthenes)

3) Print smallest prime factor for 2 to N

4) Prime factorization

5) Get the no of factors | divisors

Prime Number :- Nos having exactly 2 factors are Prime.

$$N \checkmark \to 1,\ N$$

Eg $N = 1$ ✗

Eg $N = 2$ ✓

3, 5, 7, 11, ...

Q

```
bool checkPrime(int N) {
    int cnt = 0
    for (int i = 1; i <= √N; i++) {        → i*i <= N ✓
        if (N % i == 0) {
            if (i == N/i) { cnt += 1; }
            else cnt += 2
        }
    }
    if (cnt == 2) { return true }
    else return false,
}
```

$\boxed{T.C : O(\sqrt{N})}$

$i * j = \underline{N}$, $j = N/i$ ✓

$N = 12$

i		N/i	
1		12	+2
2	<	6	+2
3		4	+2
4		3	+6 ✓
6	>	2	
12		1	

$i <= \frac{N}{i}$

$i^2 <= N$

$\boxed{i <= \sqrt{N}}$

$N = 36$ ✓

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

Q) Given a no N. Find all primes from 1 to N.

N = 10 : [1 10] : 2, 3, 5, 7 ✓

N = 20 : [1 20] : 2, 3, 5, 7, 11, 13, 17, 19

BF: for every no from 1 to N, check prime or not.

```
for(int i=1; i<=N; i++){
    if( checkPrime( i)){      // √N
        print(i)
    }
}
```

$$\boxed{\begin{array}{l} TC : O(N\sqrt{N}) \\ SC : O(1) \end{array}}$$

N = 50

F F

$\cancel{0}$ $\cancel{1}$ 2 3 $\cancel{4}$ 5 $\cancel{6}$ 7

$\cancel{8}$ $\cancel{9}$ $\cancel{10}$ 11 $\cancel{12}$ 13 $\cancel{14}$

$\cancel{15}$ $\cancel{16}$ 17 $\cancel{18}$ 19 $\cancel{20}$ $\cancel{21}$

$\cancel{22}$ 23 $\cancel{24}$ $\cancel{25}$ $\cancel{26}$ $\cancel{27}$ $\cancel{28}$

29 $\cancel{30}$ 31 $\cancel{32}$ $\cancel{33}$ $\cancel{34}$ $\cancel{35}$

$\cancel{36}$ 37 $\cancel{38}$ $\cancel{39}$ $\cancel{40}$ 41 $\cancel{42}$

43 $\cancel{44}$ $\cancel{45}$ $\cancel{46}$ 47 $\cancel{48}$ $\cancel{49}$

$\cancel{50}$

i * i

i = 2 : 4

i = 3 : 9

i = 5 : 25

i = 7 : 49

i

2i

3i

4i

5i

⋮

N

i = 7

2 * 7 (2)

3 * 7 (3)

4 * 7 (2)

5 * 7 (5)

6 * 7 (2)

7 * 7

i

i * i

N = 100

Pseudo Code :-

Given N

```
bool p[N+1] = {T}
p[0] = F, p[1] = F
for(int i=2; i<=N; i++){
    if(P[i] == T){
        // i is prime, all multiples of i
        // are non-prime
        for(int j = 2*i; j<=N; j+=i){
            P[j] = false
        }
    }
}
```

only go till $\sqrt{N}$ $i*i<=N$

instead of Starting from $2*i$, we can Start from $i*i$

$\boxed{TC \simeq O(N)}$

$\boxed{SC : O(N)}$

i = 2	4
i = 3	9
i = 5	25
i = 7	49
i = 11	121

N ✓

i   j ✓

$\sqrt{N}$ N ✓

$(\sqrt{N}+i)$ $(\sqrt{N}+1)^2 > N$

$N+1+2\sqrt{N} > N$

i — $\sqrt{N}$

$i = \sqrt{N}+1$ $(\sqrt{N}+1)^2 > N$

i	# iterations
2	N/2
3	N/3
5	N/5
7	N/7
11	N/11
⋮	⋮
P	N/P

$$\#\text{iterations} = \frac{N}{2} + \frac{N}{3} + \frac{N}{5} + \frac{N}{7} + \dots \frac{N}{P}$$

$$N\Big[\underbrace{\frac{1}{2} + \frac{1}{3} + \frac{1}{5} + \frac{1}{7} + \frac{1}{11} + \dots \frac{1}{P}}_{\text{Sum of Reciprocals of prime}}\Big]$$

→ last $P<N$

$$N\,[\log(\log N)] ✓ \simeq \underline{\underline{N}}$$

$N = 2^{30}$ $\log_2 N = \log_2 2^{30} = 30$

$\log_2 30 = <5$

Q) Given N. Find <u>smallest prime factor</u> for all numbers from <u>2 to N</u>.

SPF

Eg N = 10

	2	3	4	5	6	7	8	9	10
SPF[ ]:	2	3	2	5	2	7	2	3	2
	✓	✓		✓		✓			

$\boxed{SPF[\,]}$ ✓

2	3	4	5	6	7
2	3	~~4~~ 2	5	~~6~~ 2	7

8	9	10	11	12	13	14
~~8~~ 2	~~9~~ 3	~~10~~ 2	11	~~12~~ 2	13	~~14~~ 2

15	16	17	18	19	20	21
~~15~~ 3	~~16~~ 2	17	~~18~~ 2	19	~~20~~ 2	~~21~~ 3

22	23	24	25
~~22~~ 2	23	~~24~~ 2	~~25~~ 5

✓

SPF[j] = min(SPF[j], i)

```
if(SPF[j] == j) {
    SPF[j] = i
}
```

Pseudo Code :-

```
int SPF[N+1];

for(int i=2; i<=N; i++){
    SPF[i] = i
}

for(int i=2; i*i<=N; i++){
    if(SPF[i] == i){ // i is prime
        for(int j = i*i; j<=N; j+=i){
            if(SPF[j] == j){        // SPF[j] = min(SPF[j],
                SPF[j] = i          //          i);
            }
        }
    }
}
return SPF;
```

$$TC \simeq O(N)$$
$$SC \simeq O(1)$$

Break for 5 min :- Prime factorization
Count No of divisors

Prime Factorization :- Any number can be represented as product of power of prime nos.

2	72
2	36
2	18
3	9
3	3
	1

$$\boxed{72 = 2^3 * 3^2}$$ ✓

No of divisors | factors $= (3+1)(2+1)$

$= 12$ ✓

How ??

$72 = 2^{3} \cdot 3^{2}$

$\hookrightarrow [3^0, 3^1, 3^2]$

$[2^0, 2^1, 2^2, 2^3]$ $\hookrightarrow$ $^3C_1 = 12$

$^4C_1$ *

$2^0 * 3^0 : 1$ ✓

$2^0 * 3^1 : 3$ ✓

$2^0 * 3^2 : 9$ ✓

$2^1 * 3^0 : 2$ ✓

$2^1 * 3^1 : 6$ ✓

$2^1 * 3^2 : 18$ ✓

$2^2 * 3^0 : 4$ ✓

$2^2 * 3^1 : 12$ ✓

$2^2 * 3^2 : 36$ ✓

$2^3 * 3^0 : 8$ ✓

$2^3 * 3^1 : 24$ ✓

$2^3 * 3^2 : 72$ ✓

$\Rightarrow 12$

$\boxed{N = 600}$ ✓

2	600
2	300
2	150
3	75
3	15
5	5
	1

$\Rightarrow 2^3 * 3^2 * 5^1$

$\Rightarrow (3+1)(2+1)(1+1)$

$(4)(3)(1)$

24 ✓

Generalization :-

$$N = p_1^{x_1} \cdot p_2^{x_2} \cdot p_3^{x_3} \cdots p_y^{x_y}$$

$$\boxed{\text{No of factors} = (x_1+1)(x_2+1)(x_3+1\ldots)\cdots(x_y+1)}$$

Q) Given N. Count no of divisors through prime factorization method.

$1 * 4 * (3) * (2) = 24$ ✓

$ans = 1 * (cnt + 1)$

$N = 360$

2	360
2	180
2	90
3	45
3	15
5	5
	1

SPF(360) = 2

cnt = ~~1~~ ~~2~~ 3

x = 3, cnt = ~~0~~ ~~1~~ 2 ✓

x = 5, cnt = ~~0~~ 1

→ SPF[N+1] ✓

```
// assume that you have built the SPF array  → N
while(N > 1) {
    x = SPF[N], cnt = 0
    while(N % x == 0) {      → log N
        N = N/x
        cnt ++
    }
    ans = ans * (cnt +1)
}
```

$$\boxed{TC : O(N + \log N)}$$

Q) Given N. For every number from 1 to N, Get No. of factors.

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

✓

Build your SPF — N { N } (2nd Que) ✓

```
int cnt[N+1]; ✓ //O/P
cnt[1] = 1

for(int i = 2; i <= N; i++){
    int tmp = i , ans = 1
    while(tmp > 1){
        x = SPF[temp], cnt = 0 ✓
        while(temp % x == 0){
            tmp = temp/x
            cnt ++
        }
        ans = ans * (cnt + 1)
    }
    cnt[i] = ans
}
return cnt
```

log N

N log N

TC : O(N + N log N) = N log N

SC : O(N) (because of SPF)

That's it :-

$\boxed{\text{Imp}}$

1) Assignments
2) Contest - 1 Reattempt is still on. ✓

Thank You :

1 2 3 4

$\boxed{N^N}$ ✓

N N-1 N-2 N-3

$\boxed{N!}$

$O(N! * N)$ ✓

N - . 1