Prime Numbers

Number having only two factors.
[1 and N itself]

Eg → 7 is a prime number.

Q1 Check if given no. is prime or not?

count of factors == 2 → true.

```
boolean checkPrime( int N){
    count = 0
    for( i = 1 ; i*i <= N; i++){
        if (N % i == 0){
            if( i == N/i) { count ++ }
            else { count += 2}
        }
    }
    return (count == 2) ;
}
```

T.C → $O(\sqrt{N})$
S.C → $O(1)$

Q: Given an integer N. Check every number from 1 to N if it is a prime no.

$1 \le N \le 10^6$.

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

ans → [F T T F T F T F F F]

idea.1

```
boolean ans [N+1];

  ans[0] = ans[1] = false;

  for( i = 2 ; i <= N ; i++){
            ans[i] = checkPrime(i);
  }

  return ans[];
```

T.C → $O(N\sqrt{N})$
S.C → $O(1)$

Sieve of Eratosthenes

idea.2

0	1	2	3	4	5	6	7	8	9	10	11	12	13
F	F	T	T	F	T	F	T	F	F	F	T	F	T

14	15	16	17	18	19	20	21	22	23	24	25	26
F	F	F	T	F	T	F	F	F	T	F	F	F

2*2 = 4
2*3 = 6
2*4 = 8
|
|

3*3 = 9
3*4 = 12
|
|

5*2 = 10
5*3 = 15
5*2*2 = 20
5*5 = 25.

∴ For every prime no. (x) ⇒ first multiple to be marked as false ⇒ $x*x = \underline{x^2}$.

code....

```
boolean isPrime[N+1];    ∀i , isPrime[i] = true;

isPrime[0] = false,   isPrime[1] = false;

for( i = 2; i*i ≤ N; i++){
        if(isPrime[i] == true){
                for( j = i*i ; j <= N ; j += i){
                        isPrime[j] = false;
                }
        }
}

return isPrime[];
```

Mark all multiples of i as non-prime starting from ith multiple of i

$$\boxed{\begin{array}{l} T.C \to O(N \log(\log(N))) \\ S.C \to O(1) \end{array}}$$

i	j	iterations
i=2	4,6,8,10,12, ---N	N/2
i=3	9,12,15,18,21,24,—N	N/3
i=4	0	0
i=5	→	N/5
i=6	0	0
7	→	N/7

$$\# \text{iterations} : \left(\frac{N}{2} + \frac{N}{3} + \frac{N}{5} + \frac{N}{7} + \frac{N}{11} + \frac{N}{13} + \frac{N}{17} + \text{---} \right)$$

$$= N \left[\frac{1}{2} + \frac{1}{3} + \frac{1}{5} + \frac{1}{7} + \frac{1}{11} + \frac{1}{13} + \frac{1}{17} + \text{---} \right]$$

$$= N * \sum \frac{1}{\text{prime no.}} \Rightarrow N \cdot \log(\log(N))$$

$$\frac{1}{2} + \frac{1}{3} + \frac{1}{5} + \frac{1}{7} + \frac{1}{9} + \frac{1}{11} + \text{---} < \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \frac{1}{5} + \frac{1}{6} + \frac{1}{7} + \frac{1}{8} + \frac{1}{9} + \text{---}$$

$$\int_2^N \frac{1}{x} \cdot dx = \log x \Big|_2^N$$

$$\approx \log N$$

$$N = 2^{32}$$

$$\log_2 N = \log_2 2^{32} = 32$$

$$\log_2(\log_2 N) = \log_2 32 = \underline{\underline{5}}.$$

Q. Given an integer N.
For all the no's from 1 to N. Count of factors of the numbers.

N=10	0	1	2	3	4	5	6	7	8	9	10
ans,	[-	1	2	2	3	2	4	2	4	3	4]

idea-1. For every number, find count of factors.

[T.C → O(N√N)
S.C → O(1)]

idea-2 [Similar to sieve]

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
x	1	~~1~~	~~1~~	~~1~~	~~1~~	~~1~~	~~1~~	~~1~~	~~1~~	~~1~~	~~1~~	~~1~~	~~1~~	~~1~~	~~1~~	~~1~~	~~1~~	~~1~~
		2	~~1~~	~~2~~	2	~~2~~	2	~~2~~	~~2~~	~~2~~	2	~~2~~	2	~~2~~	~~2~~	~~2~~	2	~~2~~
			2	3		~~3~~		~~3~~	3	~~3~~		~~3~~		~~3~~	~~3~~	~~3~~		~~3~~
						4		4		4		~~4~~		4	4	~~4~~		~~4~~
												~~5~~				5		~~5~~
												6						6.

code.→

```
cf [N+1];
for( i=1 ; i <= N ; i++) {
        for( j=i ; j <= N ; j+=i) {
                cf[j] += 1 ;
        }
}
return cf [] ;
```

i	j	iterations
1	1,2,3,4,5 - - N	N
2	2,4,6,8,10,12, - - N	N/2
3	3,6,9,12,15, - -	N/3
4	→	N/4

[T.C → O(N log N) , S.C → O(1)]

Q: Given an integer N,

Find the smallest prime factor for all numbers from 2 to N.

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

ans[] → [- 2 3 2 5 2 7 2 3 2]

0	1	2	3	4	5	6	7	8	9	10
×	×	2	3	2	5	2	7	2	3	2

11	12	13	14	15	16	17	18	19	20
11	2	13	2	3	2	17	2	19	2

code →

```
spf[N+1];
for(i=2; i<=N; i++){
        spf[i] = i;
}
for( i=2; i*i <= N; i++){
        if( spf[i] == i) {   //i → prime number
            for( j= i*i ; j <=N; j+= i){
                if( spf[j] == j ) { spf[j]= i }
            }
        }
}
return spf[];
```

T.C → $O(N\log(\log(N)))$

S.C → $O(1)$

Q1) Count of factors of N. [Given spf[N+1]]

45 → 1, 3, 5, 9, 15, 45   ans → 6.

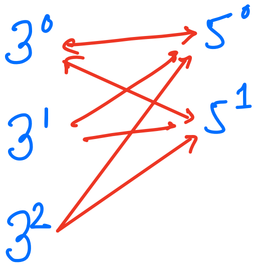
32 → 1, 2, 4, 8, 16, 32   ans → 6

45.

$\Downarrow$

$\underline{3^2 \times 5^1}$

$$\begin{array}{cccccc} \underline{1} & 5 & 3 & 15 & 9 & 45 \\ \Uparrow & \Uparrow & \Uparrow & \Uparrow & \Uparrow & \Uparrow \\ 3^0 \times 5^0 & 3^0 \times 5^1 & 3^1 \times 5^0 & 3^1 \times 5^1 & 3^2 \times 5^0 & 3^2 \times 5^1 \end{array}$$



#count of factors of $3^x * 5^y = (x+1) * (y+1)$

#count of factors of $2^a * 3^b * 5^c = (a+1) * (b+1) * (c+1)$

count of factors of $\left[600 \Rightarrow \underline{2} \times \underline{3} \times \underline{2} \times \underline{5} \times \underline{2} \times \underline{5} \right.$

$\left. \qquad \Rightarrow 2^3 \times 3^1 * 5^2 \right]$

$\Rightarrow (3+1) * (1+1) * (2+1) = 4 * 2 * 3 = \underline{24}.$

$N = 490. \Rightarrow 2^1 \times 5^1 \times 7^2$

$$\#\ \text{factors of } 490 = (1+1) * (1+1) * (2+1) = 2 * 2 * 3 = 12.$$

$$N \rightarrow P_1^a \, P_2^b \, P_3^c \, P_4^d \, --- \, P_x^x$$

$$\#\ \text{factors of } N = (a+1) * (b+1) * (c+1) * (d+1) * \text{—} (x+1)$$

$490 \xrightarrow{/2} 245$ [1]

spf[490]

$245 \xrightarrow{/5} 49$ [1]

spf[245]

$49 \xrightarrow{/7} 7 \xrightarrow{/7} 1$ [2]

spf[49]

$N=1200$

$1200 \xrightarrow{/2} 600 \xrightarrow{/2} 300 \xrightarrow{/2} 150 \xrightarrow{/2} 75$ $\boxed{\text{4 times}}$ 2^4

spf[1200]

$75 \xrightarrow{/3} 25$ $\boxed{\text{1 time}}$ 3^1

spf[75]

$25 \xrightarrow{/5} 5 \xrightarrow{/5} 1$ $\boxed{\text{2 times}}$ 5^2

spf[25]

$$\#\,\text{count of factors} = (4+1) * (1+1) * (2+1)$$

$$= 5 * 2 * 3 = \underline{30}$$

code →

```
ans = 1
while( N > 1){
      s = spf[N];   count = 0
      while( N % s == 0){
            N = N/s
            count ++;
      }
      ans = ans * (count +1);
}
return ans;
```

$$\boxed{\begin{array}{l} T.C \to O(\log_2 N) \\ S.C \to O(1) \end{array}}$$

if there are multiple queries →

then this is helpful.

$\underline{N\log N + Q.}$

$\boxed{N\log\log N} + Q\cdot\log N$

Next permutation

arr[] → [1 2 3 4 5]

1 2 3 5 4

arr[] → [3 4 5 2 7 6]

[3 4 5 6 2 7]

arr[] → [3 4 5 2 8 7 6]

[3 4 5 6 2 7 8]

arr[] → [3 4 5 $\overset{6}{\cancel{2}}$ 9 8 7 $\overset{2}{\cancel{6}}$]

[3 4 5 6 2 7 8 9]

arr[] → [9 8 7 6]. ⇒ [6 7 8 9]

arr[] → $[\ 3\quad 2\quad 1\quad 5\quad \overset{8}{\cancel{7}}\quad 9\quad \overset{7}{\cancel{8}}\quad 6\ ]$

0 1 2 3 4 5 6 7

$$\underrightarrow{[\ 3\quad 2\quad 1\quad 5\quad 8\quad 6\quad 7\quad 9\ ]}$$

① find the dip starting from r.h.s

② find the just greater number than dip on its r.h.s

③ swap dip with just greater number.

④ Reverse the subarray present on r.h.s of the dip.

- → Permutation & Combination
- → Modular arithmetic