

01 Print divisors of a number

$$N = 36$$

[1, 2, 3, 4, 6, 9, 12, 18, 36]

$$\begin{aligned} TC &= O(\sqrt{n}) \\ SC &= O(1) \end{aligned}$$

$$1 \times 36 = 36$$

$$2 \times 18 = 36$$

$$3 \times 12 = 36$$

$$4 \times 9 = 36$$

$$6 \times 6 = 36$$

Both all
divisors.

02 Check if a number is prime or not

No is divisible by itself and by 1 and no. of divisors = 2

$$cnt = 0$$

$$\text{for } (i = 1 - \sqrt{n}) \{$$

$$\text{if } (N \% i == 0) \{$$

$$cnt = cnt + 1$$

$$\text{if } (n / i != i) \{$$

$$cnt = cnt + 1$$

}

$$\begin{aligned} TC &= O(\sqrt{n}) \\ SC &= O(1) \end{aligned}$$

03 Prime factors of given N.

Ex. $N = 60$

1, 2, 3, 4, 5, 6, 10

12, 15, 20, 30, 60

Ex. $N = 30$

5, 7

Ex. $N = 780$

2, 3

04 Power exponentiation • Brute Force

$$x = 2$$

$$n = 5$$

Now, $\text{Power}(2, 5)$
 $= 32.$

Now $x = 2$ and $n = 10$
 $2^4 \rightarrow (2^2)^2 \rightarrow (2^2)^{2/2} = (4)^1$
 $\text{ans} = 1 \times 2$

reduction possible.

Now, $4^{10} \rightarrow (4^2)^{10/2}$
 $= (16)^5$

Now, $16^8 \rightarrow (16) \times 16^4$
 $\text{ans} = 1 \times 2 \times 16$

$16^4 = ((16)^2)^{4/2}$
 $= (256)^2$

$(256)^2 = (256)^2^{2/2}$
 $= (65536)^1$

So, $\boxed{\text{ans} = 1 \times 2 \times 16 \times 65536}$

- When power is odd, remove and add it to simply in ans.

run loop and multiply again and again.

$\text{Power}(x, n) \{$

$\text{ans} = 1$

while $(n > 0) \{$

if $(n \% 2 == 1) \{$

$\text{ans} = \text{ans} \times x;$

$n = n - 1;$

else $\{$

$n = n / 2;$

$x = x \times x;$

$\}$ return ans;

Now $5^{-2} = 1/25$

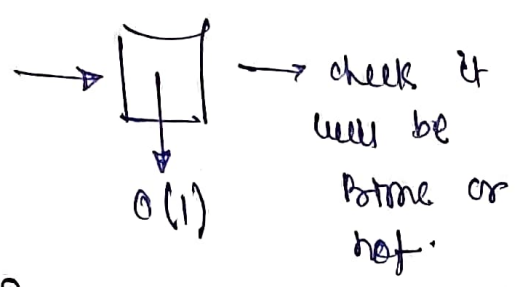
→ simply compute they and return $(1/25)$.

- if we have to handle double value. then put double.

$Tc = O(\log_2 n)$
 $Sc = O(1).$

Q9 Sieve of Eratosthenes

Given a no. N , print all primes from till N .



Q. $N=30$ Prime [31]

2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
1	1	0	1	0	1	0	0	0	1	0	1	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	1	0

Here all are marked with 0 and 1, 0 should not prime one (1) will show these all primes.

Prime(N) {

Prime [N+1] \rightarrow (2 \rightarrow N) as 1

for (i = 2 \rightarrow N) Prime [i] = 1

for (i = 2 \rightarrow N) { \sqrt{N}

if (Prime [i] == 1) {

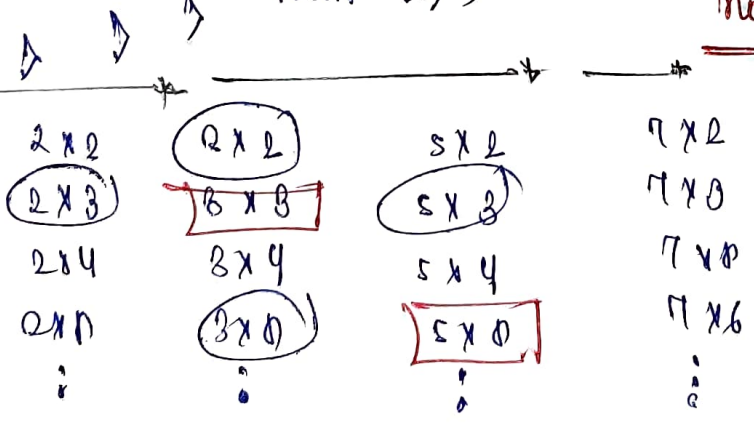
for (j = 2 * i; j <= N; j += i) {
Prime [j] = 0;

for (i = 2 \rightarrow N) {

if (Prime [i] == 1) {

Print (i);

now optimized



$$TC = O(N \log(\log N)) + O(N) + O(N)$$

$$SC = O(N)$$