Omansh Mathur
omansh.mathur_1@
scaler.com

Q1) Given a number, find if it is prime.

Prime number → Number with only 2 factors. 1 & itself.
Eg - 3, 5, 7

23 is a prime.

Q Is 1 prime.

Factors → 1

Idea : Count the number of factors

```
bool is_prime(int n)
{
    int count = 0
    for(int i=1; i<=n; i++)
    {
        if(n%i == 0)
            count++;
```

```
}
if (count == 2)
    return true;
else
    return false;
}
```

Iterations = n

Eg 1 21

i = 1 count = 1

i = 2

i = 3 count = 2

4

5

6

7 count = 3

.

.

.

i = 21 count = 4

Eg 2 23 count = 2

i = 1

i = 23

Eg 3 $n = 1$

$i = 1$ count = 1

Assumption → 1 sec 10^8 op

$n = 10^9 \longrightarrow \frac{10^9}{10^8} = 10$ sec

$10^1 \times 10^8$

1 sec 10^8

10 sec 10^9

$n = 10^{18} \longrightarrow \frac{10^{18}}{10^8} = 10^{10}$ sec

$\simeq 317$ yrs

Obs: If $a * b = N$ then the pair (a, N/a) both are factors.

$b = N/a$

$N = 24$

i	N/i
1	24
2	12
3	8
4	6
6	4
8	3
12	2
24	1

$a \leq 60$

$max_a = ? \; 60$

$i \leq N/i$

$i_{max} = N/i_{max} \Rightarrow i_{max}^2 = N$

$i_{max} = \sqrt{N}$

```
bool is_prime(int n)
{
    int count = 0
    for(int i=1; i*i<=n; i++)
    {
        if(n%i==0)
        {
            if(i*i==n)
                count+=1
```

```
        else
            count += 2
        }
    }
    if (count == 2)
        return true
    else
        return false
}
```

n = 9

{1, 3, 9}

Iterations = $\sqrt{N}$

n = 36

i = 1
2
3
4
5
6

count
2
4
6
8
8
9

n = 23

i = 1

count
2

2	2
3	2
4	2

$n = 10^{18}$ iterations $= 10^9$

$= 10$ sec

Carl Friedrich Gauss

Find the sum of integers from 1 to 100

$S = 1 + 2 + 3 + \cdots + 99 + 100$

$S = 100 + 99 + 98 + \cdots + 2 + 1$

$2S = \underbrace{101 + 101 + 101 + \cdots + 101}_{100 \text{ times.}}$

$2S = 101 \times 100$

$$S = \frac{101 \times 100}{2} = 5050$$

$$S = 1 + 2 + 3 + \cdots + n$$

$$S = n + n-1 + \cdots + 1$$

$$2S = \underbrace{(n+1) + (n+1) + \cdots + (n+1)}_{n \text{ times.}}$$

$$2S = n(n+1)$$

$$S = \frac{n(n+1)}{2}$$

Q Given a number n, how many times can you divide by 2 till it reaches 1.

(integer division)

$7/2 = 3$

2^0 1 $\longrightarrow$ 0

2^1 2 $\longrightarrow$ 1 1

2^2 4 $\longrightarrow$ 2 $\longrightarrow$ 1 2

2^3 8 $\longrightarrow$ 4 $\longrightarrow$ 2 $\longrightarrow$ 1 3

2^4 16 $\longrightarrow$ 8 $\longrightarrow$ 4 $\longrightarrow$ 2 $\longrightarrow$ 1 4

2^5 32 $\longrightarrow$ 16 $\longrightarrow$ 8 $\longrightarrow$ 4 $\longrightarrow$ 2 $\longrightarrow$ 1 5

2^6 64 $\longrightarrow$ 32 $\longrightarrow$ 16 $\longrightarrow$ 8 $\longrightarrow$ 4 $\longrightarrow$ 2 $\longrightarrow$ 1 6

51 $\longrightarrow$ 25 $\longrightarrow$ 12 $\longrightarrow$ 6 $\longrightarrow$ 3 $\longrightarrow$ 1 5

60 $\longrightarrow$ 30 $\longrightarrow$ 15 $\longrightarrow$ 7 $\longrightarrow$ 3 $\longrightarrow$ 1 5

32 33 34 ---- 51 ... 60 ... 63

$n \xrightarrow{\text{ans}} y$

y is called the logarithm of n.

$\log_2 n = y$

$\log_2 64 = 6$

$\log_2 51 = 5$

$\log_3 n$ → no of times you can divide by 3 till it reaches 1

```
int log ( int n)
{   int count =0;
    while (n != 1)
    {
        n = n/2;
        count ++;
    }
    return count;
}
```

n = 64
    32

16	2
8	3
4	4
2	5
1	6

50	
25	1
12	2
6	3
3	4
1	5

Break

$\sqrt{n}$

Back at 10:35

$\log(n)$

Q Amazon

Given a perfect square no, find the square root of the number.

Eg 25 → 5

49 → 7

N is perfect - square means that an integer x exists st

$$x * x = N$$

Idea: Square root will lie between 1 & N.

```
int squareroot ( int n)
{
    for ( int i = 1; i <  ; i++)
    {   if ( i*i == n)
            return i;
    }
}
```

n = 36

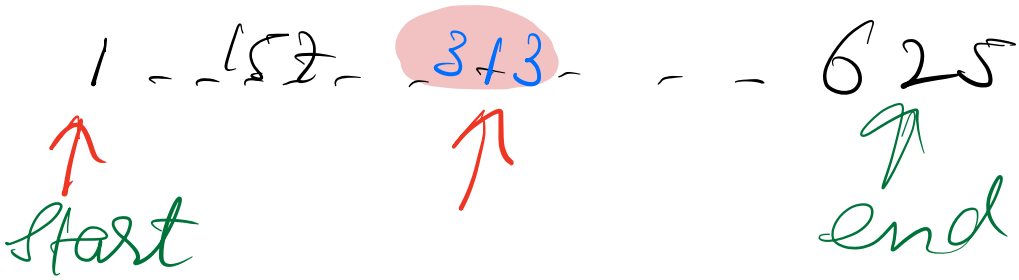
i = 1
i = 2
i = 3
i = 4
i = 5
i = 6 ←

Number of iterations $= \sqrt{N}$

Binary search.

Obs: If n increases, the squareroot also increases.

625



$$mid = \frac{(1+625)}{2} = 313$$

$313 * 313 = 625$ X

$313 * 313 > 625$

$$mid = \frac{(1+313)}{2} = 157$$

$$152 * 152 > 625$$

$$\left(\frac{1 + 152}{2}\right) = 79$$

$$79 * 79 > 625$$

$$\frac{1+79}{2} = 40$$

$$40 * 40 > 625$$

$$\left.\frac{1+40}{2}\right] = 20 \quad 1 \quad 40$$

$$20 * 20 < 625$$

$$20 \quad 40$$

$$20 \quad 40$$

$\left(\frac{20+40}{2}\right) = 30$

$30 * 30 > 625$

20 30

$\frac{20+30}{2} = 25$

$25 * 25 = 625$

1 . . . . . 625 625

1 313 313

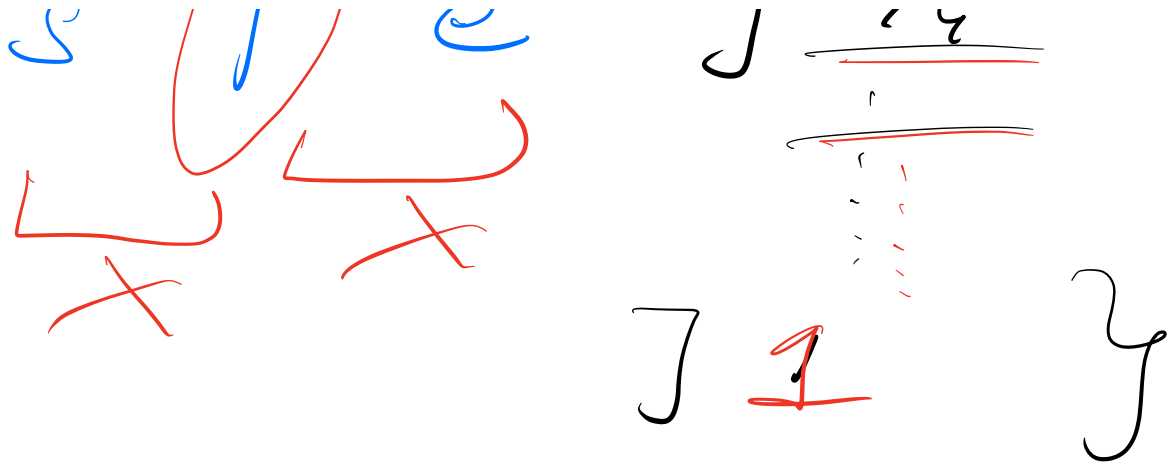
1 157 157

1 79 79

1 - - n $\int n$ no.s

$\int \frac{n/2}{}$

$\int n/1$



Number of iterations in binary search = $\log_2 n$

$n = 10^{18}$

$\log_2 10^{18} \approx 60$

$\sqrt{10^{18}} = 10^9$

end != start

while (start <= end)

{ mid = start + end

$10^8 \rightarrow 1$ sec

10^9

$= 10 \times 10^8 \rightarrow 10$ sec

$10^8 \rightarrow 1$

$10 \rightarrow 1/10^8 \times 10 = 10^{-7}$

$\sqrt{n} \rightarrow x$

$x * x = n$

$\log(n) \rightarrow \boxed{x}$

$n \rightarrow n/2 \rightarrow n/4 \rightarrow \ldots \rightarrow 1$

x times.