1) Intro
2) Get all prime no. from 1 to N.
3) Print smallest prime factor (SPF) from 2 to N.
4) Prime factorisation
5) Get the no. of factors / divisors.

Prime Number

Number having exactly 2 factors : 1 & itself.

Eg: 2, 3, 5, 7, 11, 13.....

Given an integer N. Check if it is prime or not.

Soln

Count factors
if (count == 2) { prime }

Given a no. N. We need to print all the prime nos from 1 to N.

N = 10 ⇒ 2, 3, 5, 7

N = 20 ⇒ 2, 3, 5, 7, 11, 13, 17, 19.

Sol^n 1) Brute force

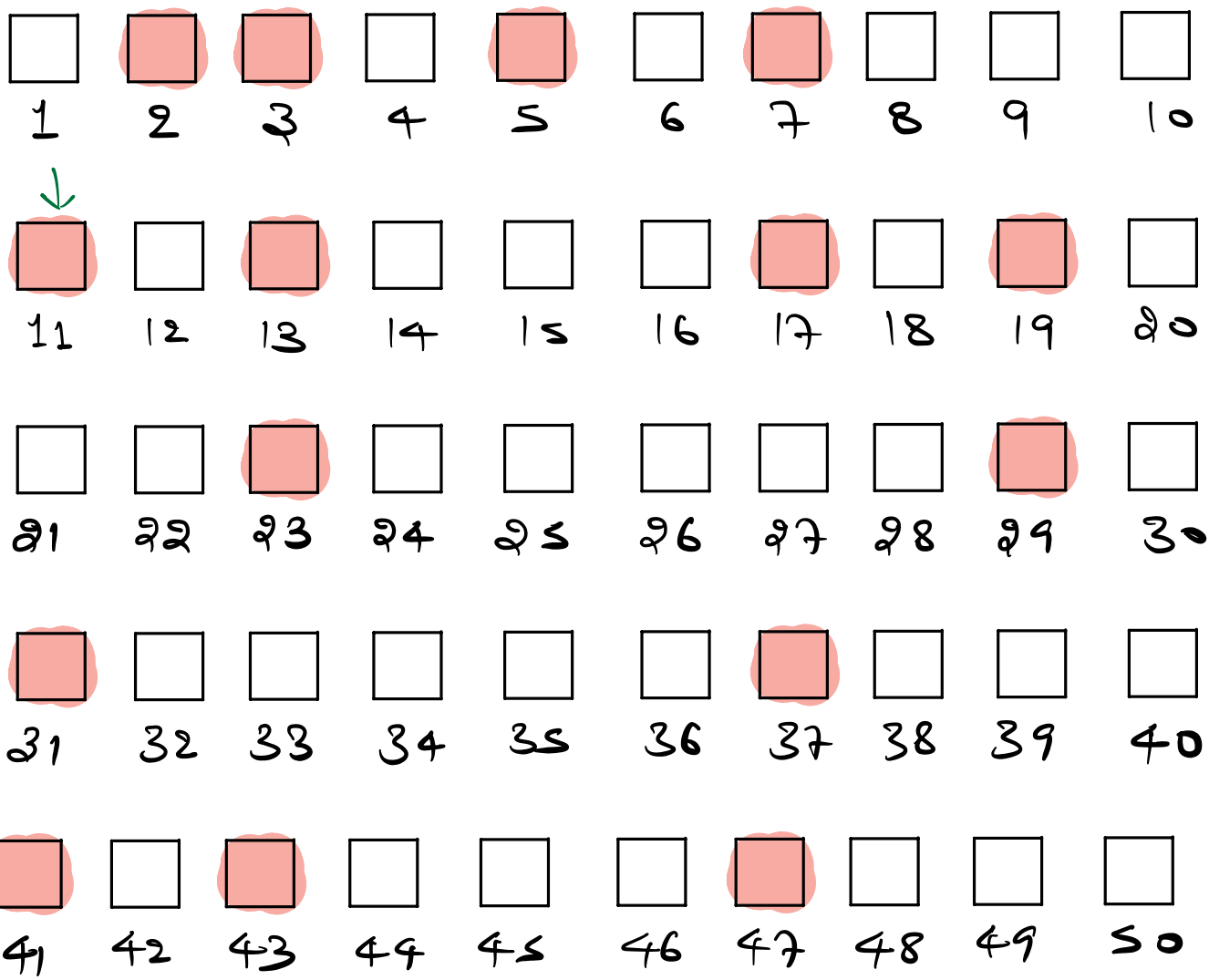
$\forall i \rightarrow [1, N]$, check if count of factors is 2

N $\quad$ $\sqrt{N}$

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

Sieve of Eratosthenes.

$N = 50$



2 ⇒ 2×2, 2×3, 2×4, 2×5 ..... 2×25.

3 ⇒ 3×2, 3×3, 3×4, 3×5 ..... <=N.
(+3, +3, +3)

5 ⇒ 5×2, 5×3, 5×4, 5×5, 5×6, 5×7 ..... <=N

7 ⇒ 7×2, 7×3, 7×4, 7×5, 7×6, 7×7, ~~7×8~~ > N.

---

11 ⇒ 11×2, 11×3 ..... 11×10, 11×11 > N

(Square of i) ≤ N

Time Complexity.

T.C. = O(N log(log(N)))

i [2 → √N]

j = [i×i → N]

2	≃ N/2
3	≃ N/3
⋮	⋮
√N	≃ 1 iteration.

$$T.C. = \frac{N}{2} + \frac{N}{3} + \frac{N}{5} + \frac{N}{7} \cdots\cdots$$

$$= N\left(\underbrace{\frac{1}{2} + \frac{1}{3} + \frac{1}{5} + \frac{1}{7} \cdots\cdots \frac{1}{\infty}}\right)$$

Sum of reciprocals of all prime no.

$$\Downarrow$$

$$T.C. = N \times \log(\log N)$$

$$\log_5(\log N) < \log_{32}(N) < \underset{(10^4/10^5)}{\sqrt{N}} \quad \left(\text{for large } N\right)$$

$$N = 2^{32} \simeq 2 \times 10^9$$

$$S.C. = \underline{O(N)}$$

Code

```
void printAllprimes (N) {

boolean isPrime[N+1] = {true};
```

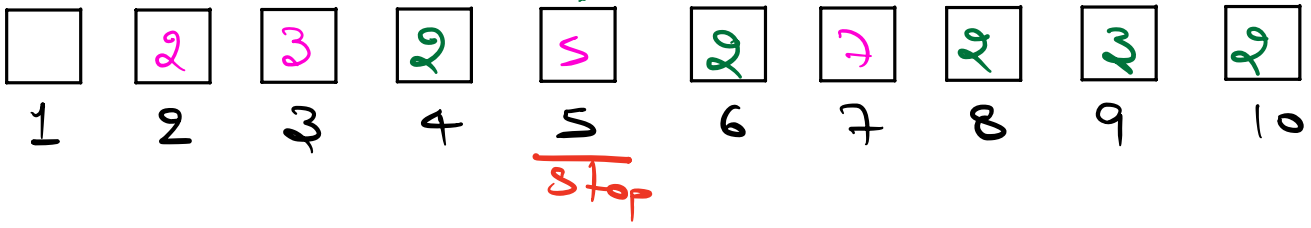
```
isPrime[0] = isPrime[1] = false;
for (i=2;  i*i <= N;  i++) {

    if (isPrime[i] == true) {

        for (j = i*i;  j <= N;  j = j+i) {

            isPrime[j] = false;

        }

    }

}

print all indexes whose value
      is true;
```

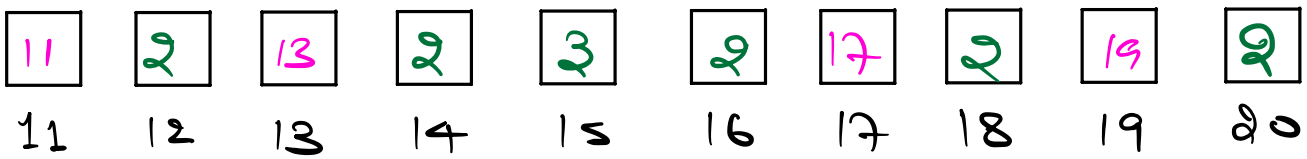
Given N. Return the SPF (smallest prime factor of all the numbers from 2 to N.

Google
MS
facebook
⋮

2, 3, 4, 5, 6, 7, 8, 9, 10

N=10 2 3 2 5 2 7 2 3 2





for a prime no. i, spf[i] = i.

Code

```
int[] spf (int N) {

    int spf[N + 1];

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

    for (i = 2; i*i <= N; i++) {
            if (spf[i] == i) {

                for (j = i*i; j <= N; j = j + i) {

                    if (spf[j] == j) {
                            spf[j] = i;
                    }

                }

            }
    }

    return spf;
```

Basic Prime factorisation

Process of finding the prime numbers which are multiplied to get the original no.

$$16 = 2 \times 2 \times 2 \times 2$$

$N = 48$

Using spf

spf	
2	48
2	24
2	12
2	6
3	3
	1

$$48 = 2 \times 2 \times 2 \times 2 \times 3$$

$$= 2^4 \times 3^1$$

divisors of 48 $= 2^{x} \times 3^{y}$

x: [0, 1, 2, 3, 4] — 5

y: [0, 1] — 2

$\underset{x}{5} \times \underset{y}{2} = 10$ divisors.

48 = 1, 2, 3, 4, 6, 8, 12, 16, 24, 48.

N = 300

2	300
2	150
3	75
5	25
5	5
	1

$300 = 2^2 \times 3^1 \times 5^2$

Count = $(2+1) \times (1+1) \times (2+1)$

$= 3 \times 2 \times 3 = 18$

Q. for a given no. N. find the count of divisors of all nos from 1 to N.

Approach

2	10000
2	5000
2	2500
2	1250
5	625
5	125
5	25
5	5
	1

4 (2's) + 4 (5's) $\leq \log_2 N$

ans = $1 \times 5 \times 5$

$= 25$

H.W.

Code

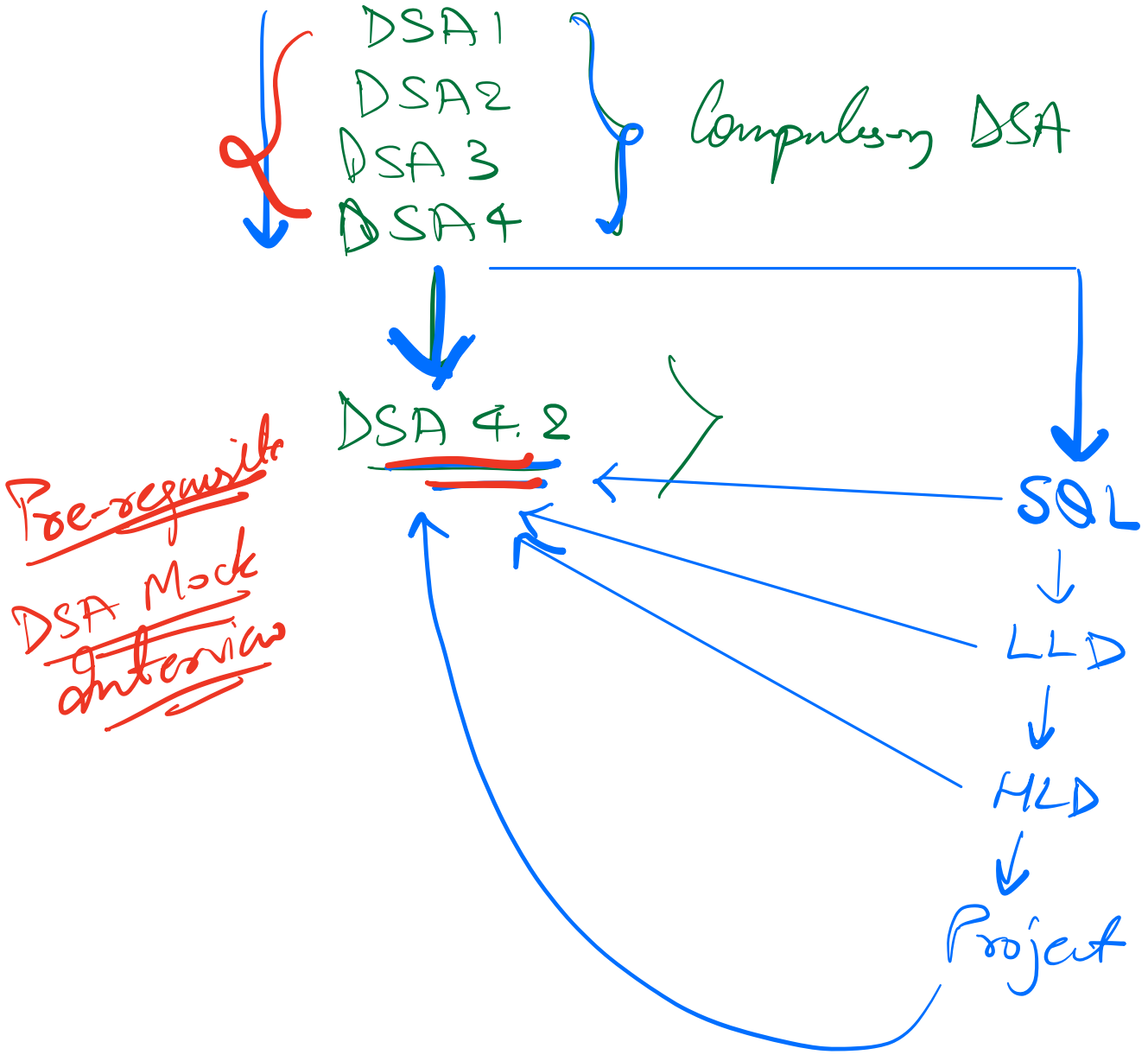
N
↓/2
N/2
↓/2
⋮
1

$\log N$

<10%

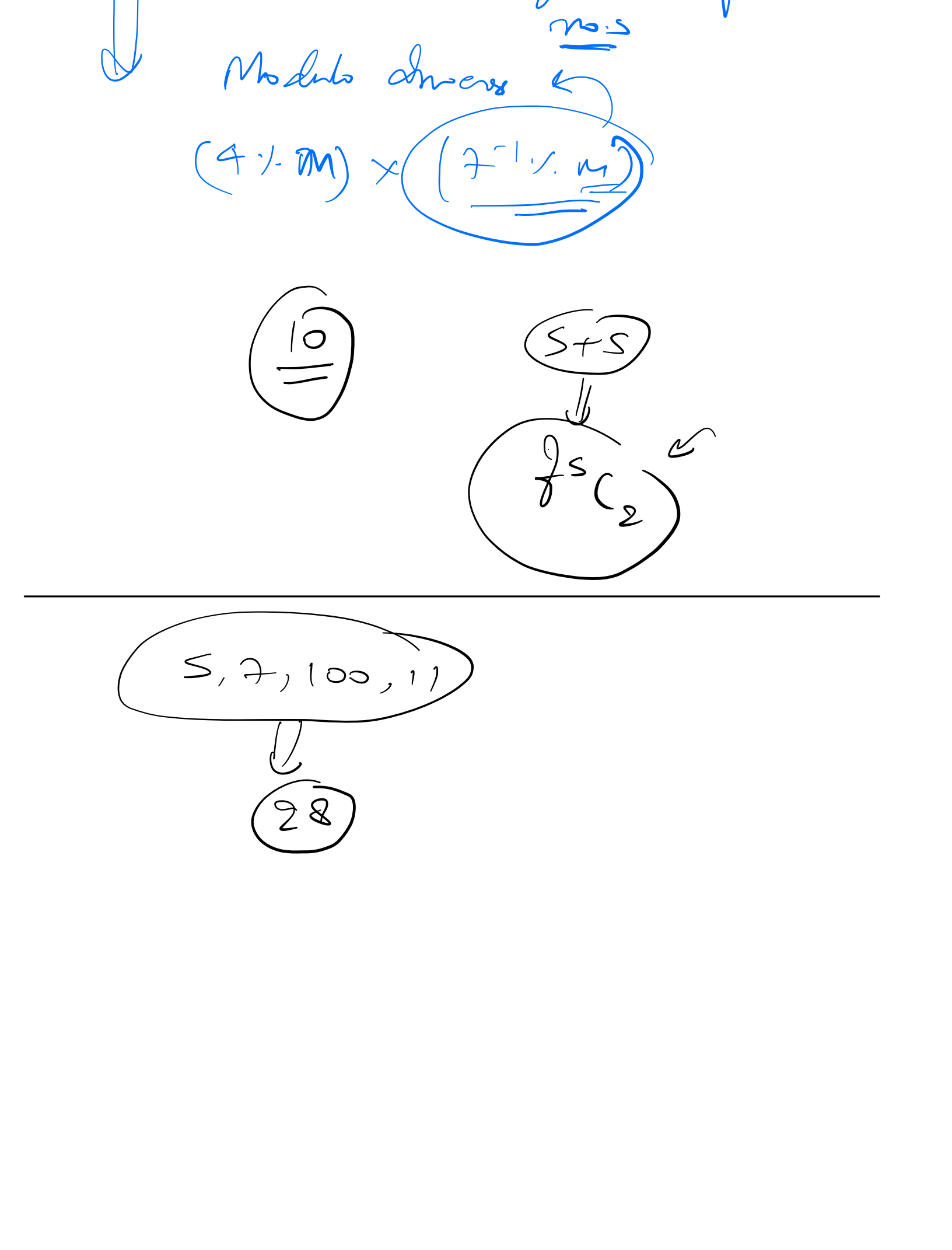
>90%

Google
Discuss
Codechef
Codeforces



$\left(\frac{4}{7}\right) \cdot M \left(\frac{9}{5}\right) \cdot M = \frac{9 \cdot M}{5 \cdot M}$

Mod is not defined for float


Modulo division
(4 % 7M) × (7⁻¹ % m)
10
5+5
$^5C_2$
5, 7, 100, 11
28