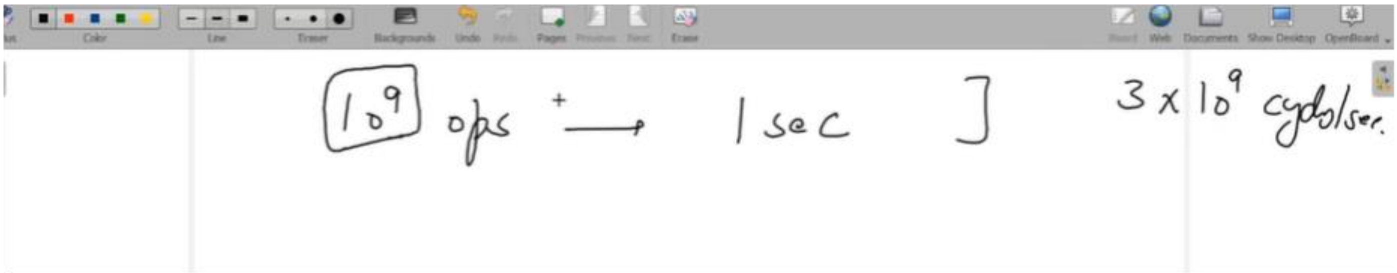
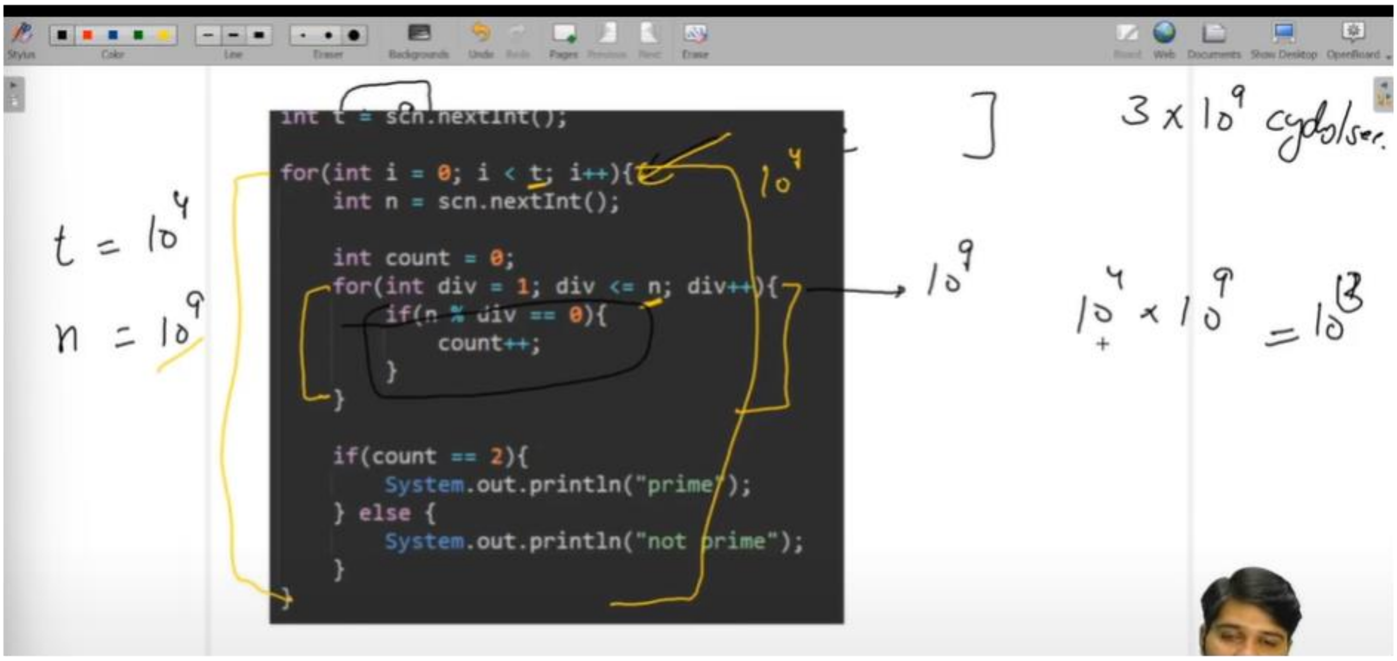


We can perform 10^9 operations in one second



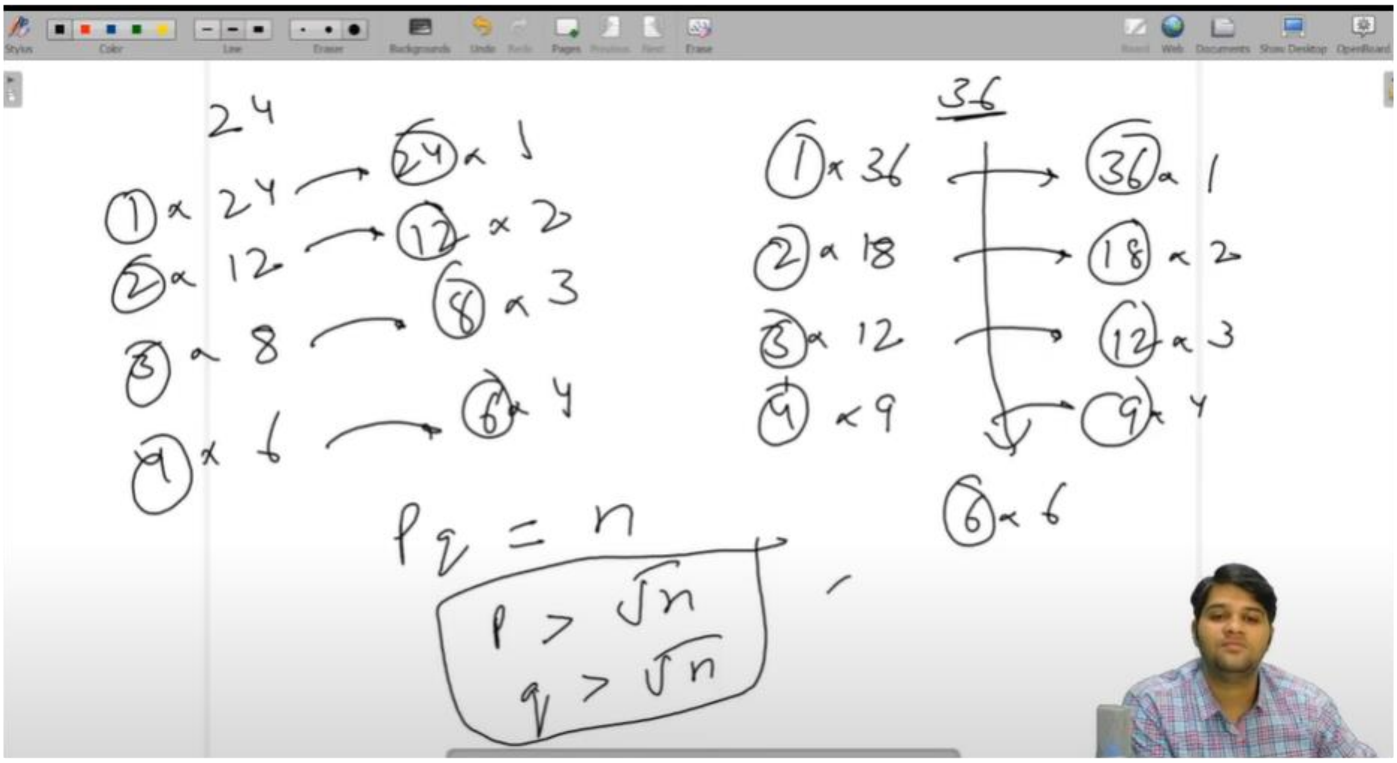
```
int t = scn.nextInt();

for(int i = 0; i < t; i++){
    int n = scn.nextInt();

    int count = 0;
    for(int div = 1; div <= n; div++){
        if(n % div == 0){
            count++;
        }
    }

    if(count == 2){
        System.out.println("prime");
    } else {
        System.out.println("not prime");
    }
}
```

NOW AS PER CODE WRITTEN IN WRONG ATTEMPT WE HAD EXCEED THE TIME LIMIT

BECAUSE OUR OUTTER LOOP RUNS 10^4 TIMES

WE NEED TO RUN INNER OPERATIONLOOP 10^9 MAX TIMES

THUS IF WE SEE THE INNER LOOP RUNS 10^13 TIMES WHICH IS GREATER THAN GIVEN OPERATIONS



🟢 **Left Side (24):**

The teacher is listing all the factor pairs of 24:

- 1 × 24
- 2 × 12
- 3 × 8
- 4 × 6

Then it stops at **5**, because:

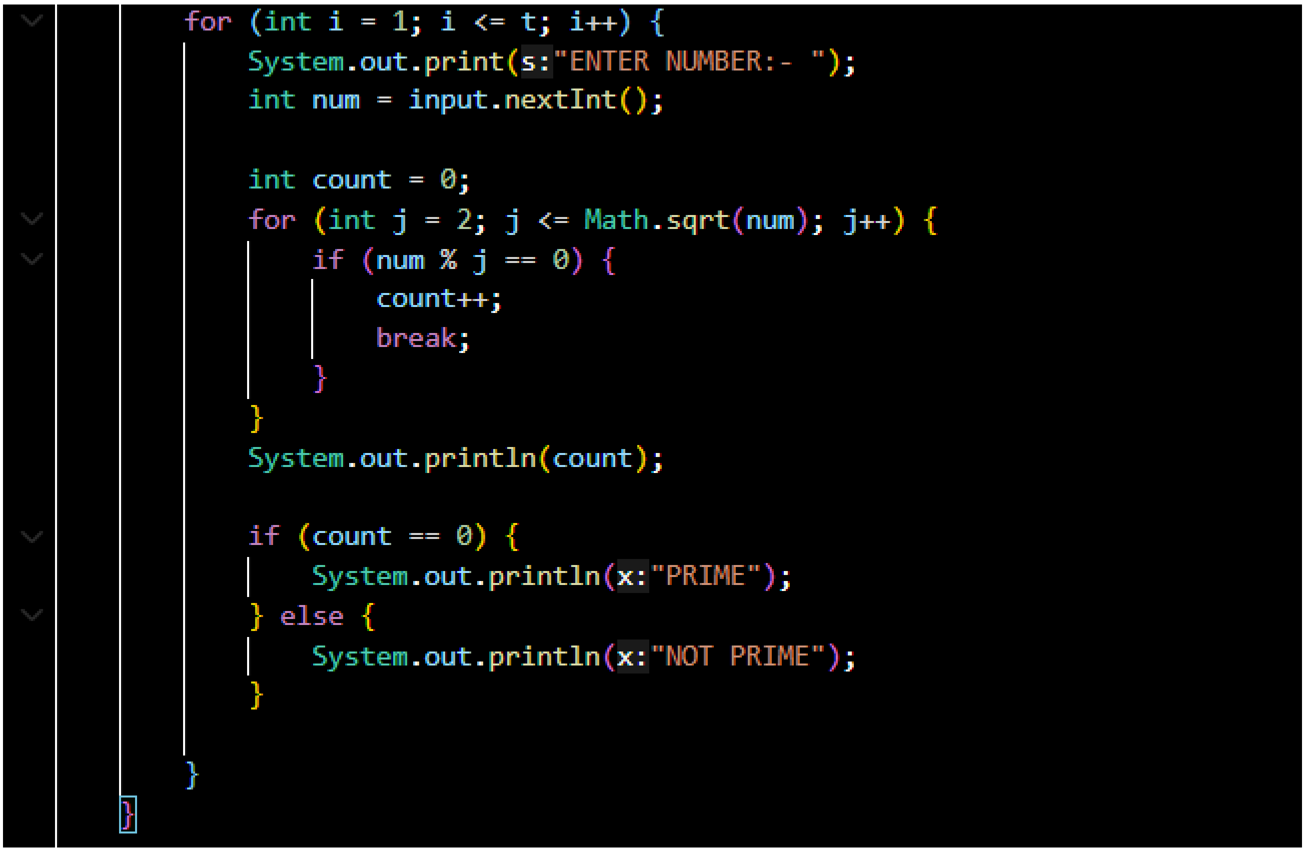
- After 4 × 6, we start repeating the same factors (like 6 × 4, 8 × 3, etc.).
- So we can **stop checking at √24 ≈ 4.9**.

💡 **Only check up to square root of the number** to save time.

🧠 **Smart Trick for Prime Check:**

If you want to check if a number n is **prime**, you only need to check for divisibility from **2 to √n**.
If no number divides n in that range, it's a prime!

OPTIMIZED SOLUTION



```
    for (int i = 1; i <= t; i++) {
        System.out.print(s:"ENTER NUMBER:- ");
        int num = input.nextInt();

        int count = 0;
        for (int j = 2; j <= Math.sqrt(num); j++) {
            if (num % j == 0) {
                count++;
                break;
            }
        }
        System.out.println(count);

        if (count == 0) {
            System.out.println(x:"PRIME");
        } else {
            System.out.println(x:"NOT PRIME");
        }

    }
}
```

LET CONSIDER OUTER LOOP TAKES 10^4 TIMES

AND INNER LOOP CAN TAKE MAX 10^9 TIMES BUT AS WE DO SQRT IT TAKES NOW 10^3 TIMES

THUS MAX IT TAKES 10^7 TIMES THUS IT DOESN’T EXCEED TIME

WORK FLOW:-

If user enters 7:

- **Loop checks: 2, 3 (since √7 ≈ 2.6)**
- **7 is not divisible by 2 or 3 → count stays 0 → PRIME**

If user enters 9:

- **Loop checks: 2, 3 (since √9 = 3)**
- **9 is divisible by 3 → count becomes 1 → NOT PRIME**