Number Theory Lecture 1

Thursday, 4 July 2024 8:19 PM

Prime Numbers

└ 2, 3, 5, 7, 11, . . .

1 — Neither prime NOR composite

① Checking Primilarity

Given a number n, check if it is prime or not,

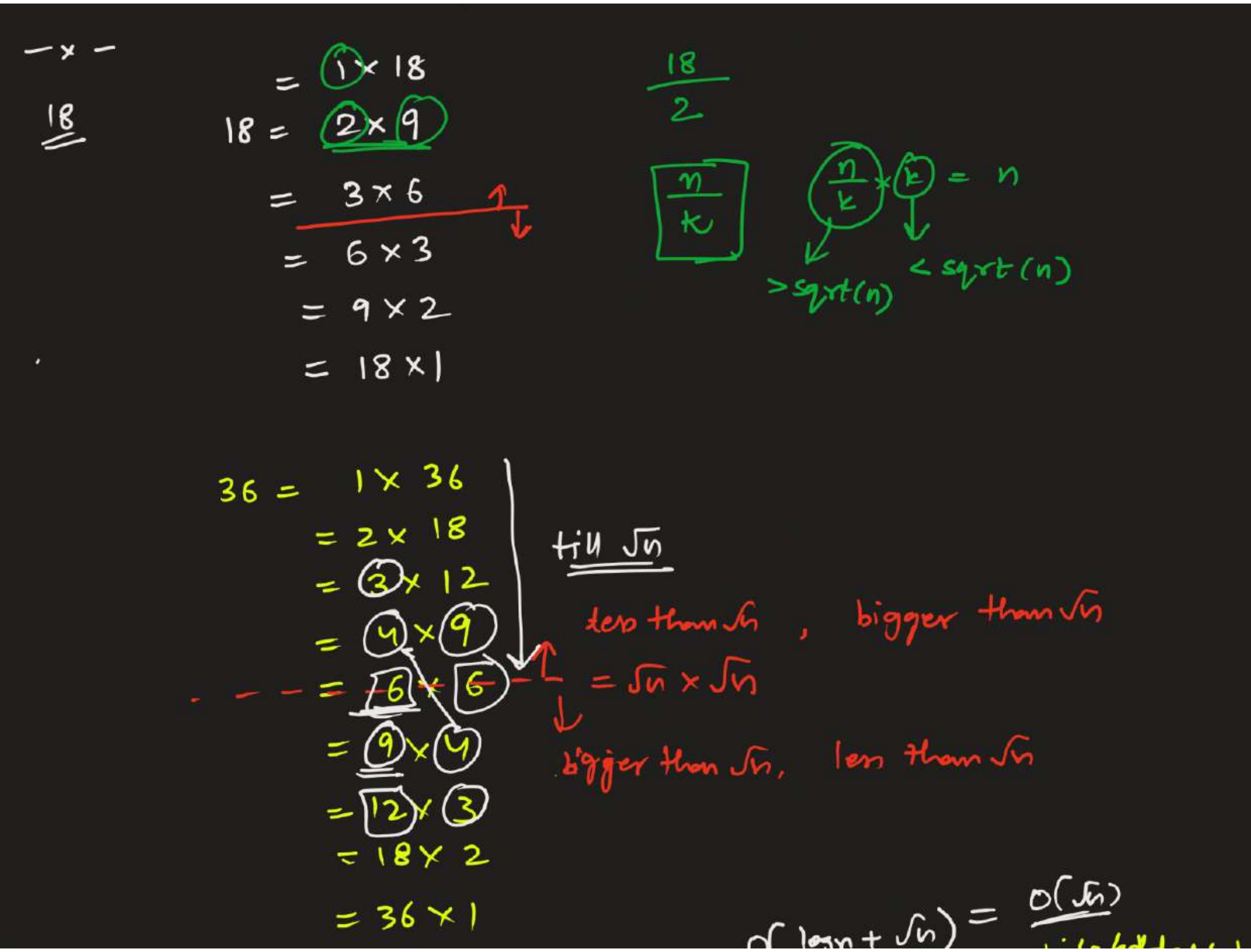
Constraints :- $1 \le n \le 10^{10}$

$O(n)$ ↓ TLE

$O(\sqrt{n})$ ✓

```
for ( i= 2 → n-1) {
    if ( n%.i = =0) {
        print "NO";
        return
    }
}
print "Yes";
```

O(N)



— x —

18

$18 = 1 \times 18$

$= 2 \times 9$

$= 3 \times 6$

$= 6 \times 3$

$= 9 \times 2$

$= 18 \times 1$

$\frac{18}{2}$

$\boxed{\frac{n}{k}}$

$\frac{n}{k} \times k = n$

$> sqrt(n)$ $< sqrt(n)$

$36 = 1 \times 36$

$= 2 \times 18$

$= 3 \times 12$

$= 4 \times 9$

$= 6 \times 6$

$= 9 \times 4$

$= 12 \times 3$

$= 18 \times 2$

$= 36 \times 1$

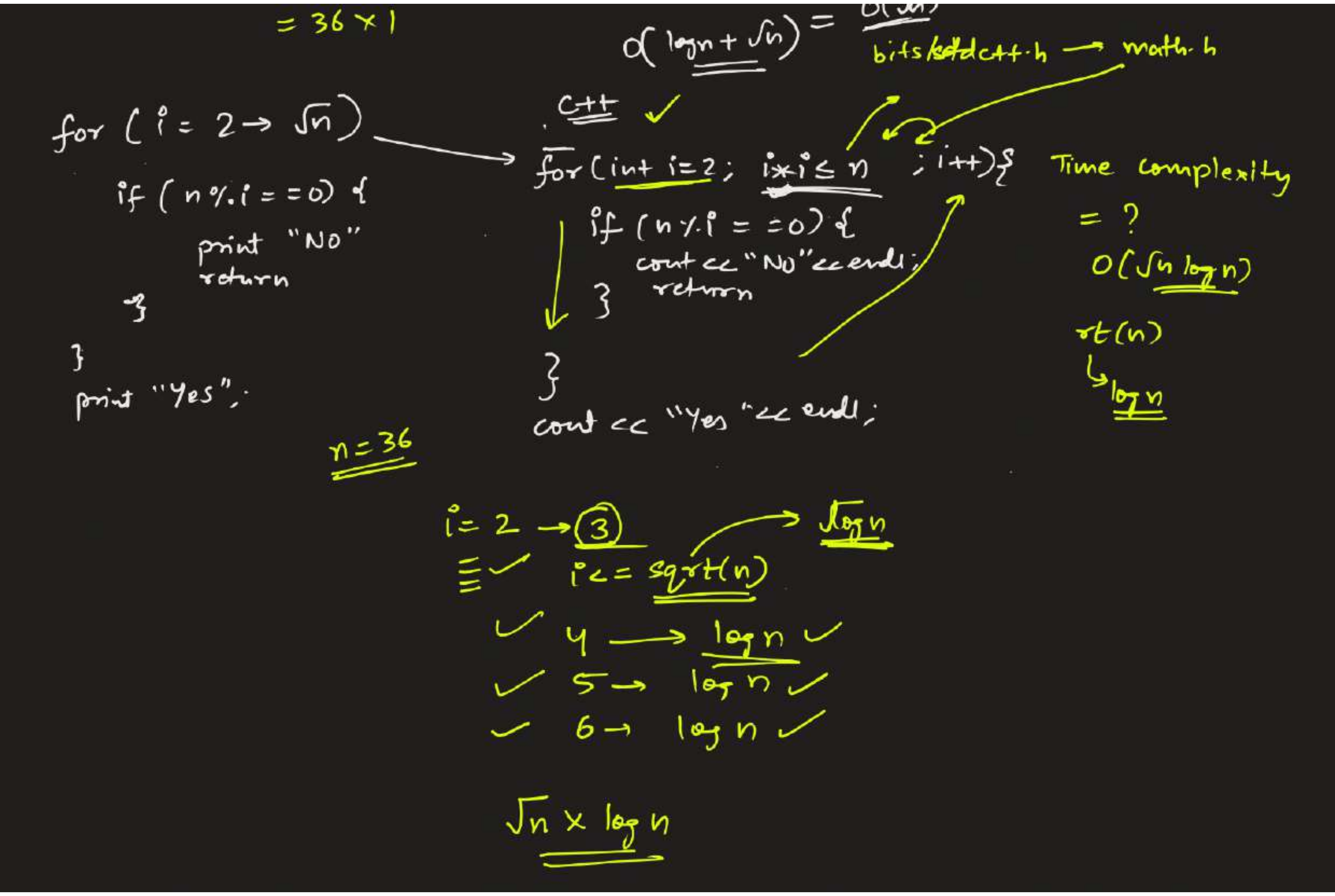
till $\sqrt{n}$

less than $\sqrt{n}$, bigger than $\sqrt{n}$

$= \sqrt{n} \times \sqrt{n}$

bigger than $\sqrt{n}$, less than $\sqrt{n}$

$O(\log n + \sqrt{n}) = O(\sqrt{n})$



$= 36 \times 1$

$O(\log n + \sqrt{n}) = O(\sqrt{n})$

bits/stdc++.h → math.h

```
for ( i = 2 → √n )
    if ( n % i = = 0) {
        print "NO"
        return
    }
}
print "Yes";
```

C++ ✓

```
for(int i=2; i*i ≤ n ; i++){
    if (n % i = = 0) {
      cout << "NO" << endl;
      return
    }
}
cout << "Yes" << endl;
```

Time complexity = ?

$O(\sqrt{n} \log n)$

sqrt(n)

↳ log n

n = 36

i = 2 → ③ → $\sqrt{\log n}$

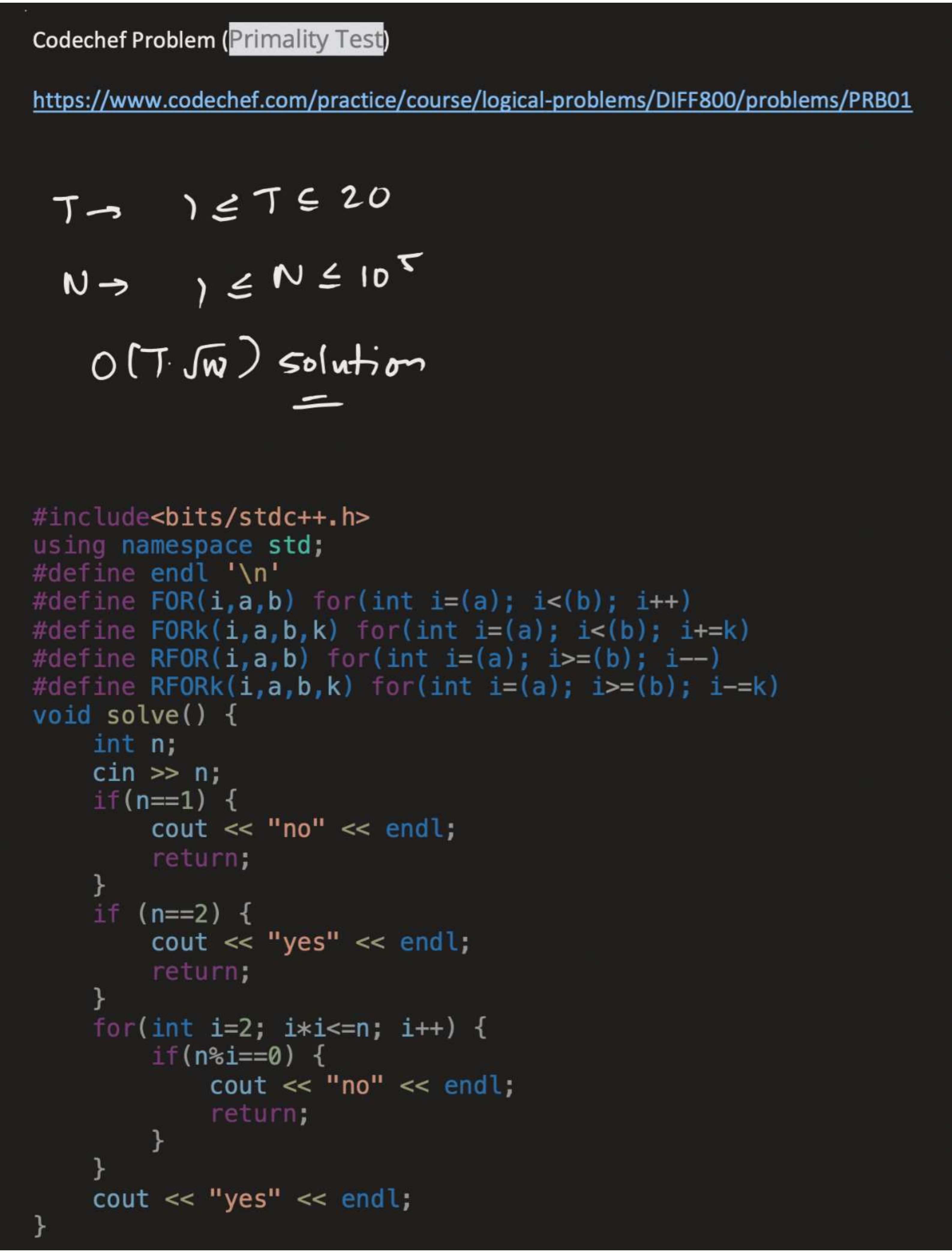
≡ ✓ i <= sqrt(n)

✓ 4 → log n ✓

✓ 5 → log n ✓

✓ 6 → log n ✓

$\sqrt{n} \times \log n$



Codechef Problem (Primality Test)

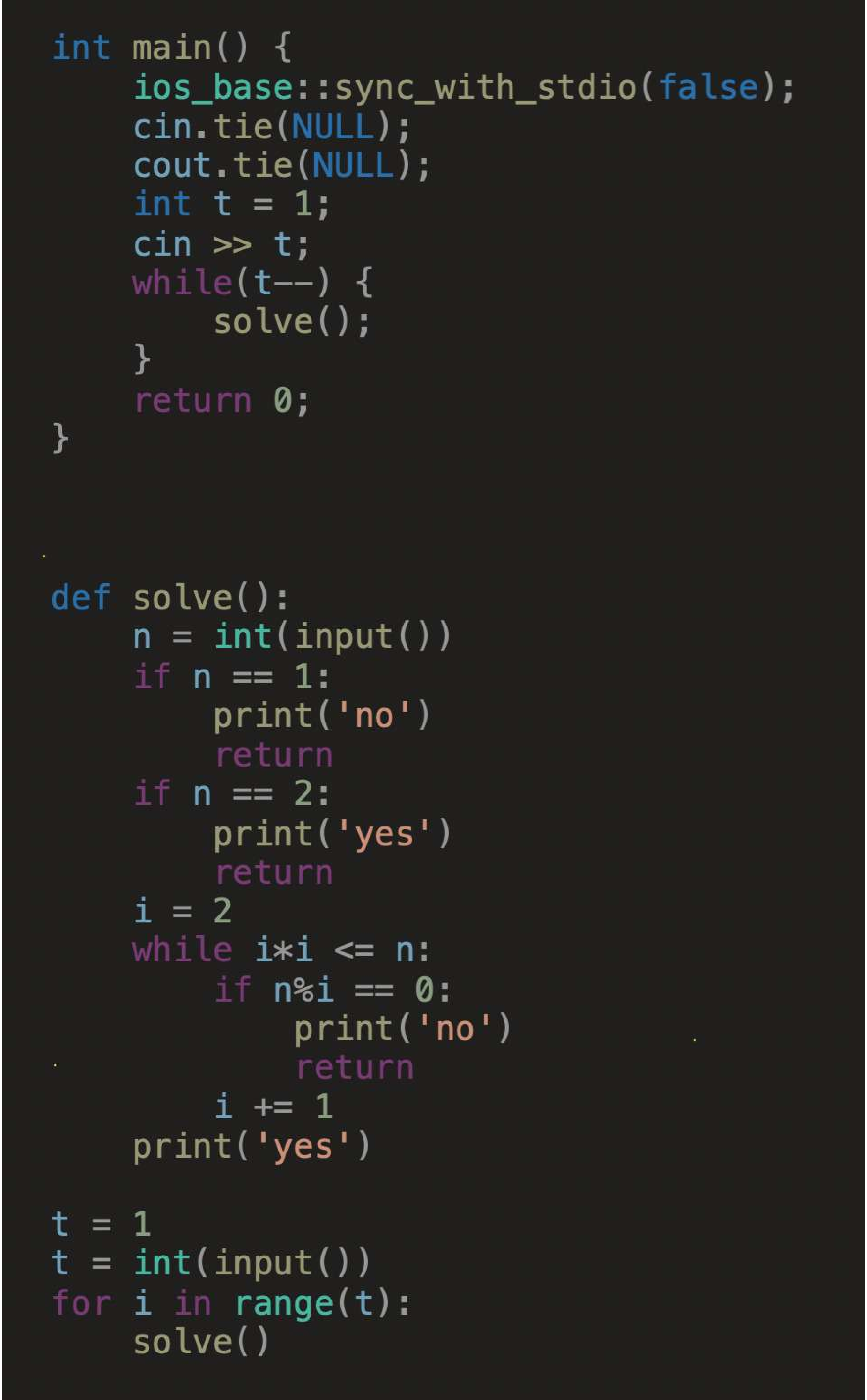
https://www.codechef.com/practice/course/logical-problems/DIFF800/problems/PRB01

T → $1 \leq T \leq 20$

N → $1 \leq N \leq 10^5$

$O(T \cdot \sqrt{N})$ $\underline{\text{solution}}$

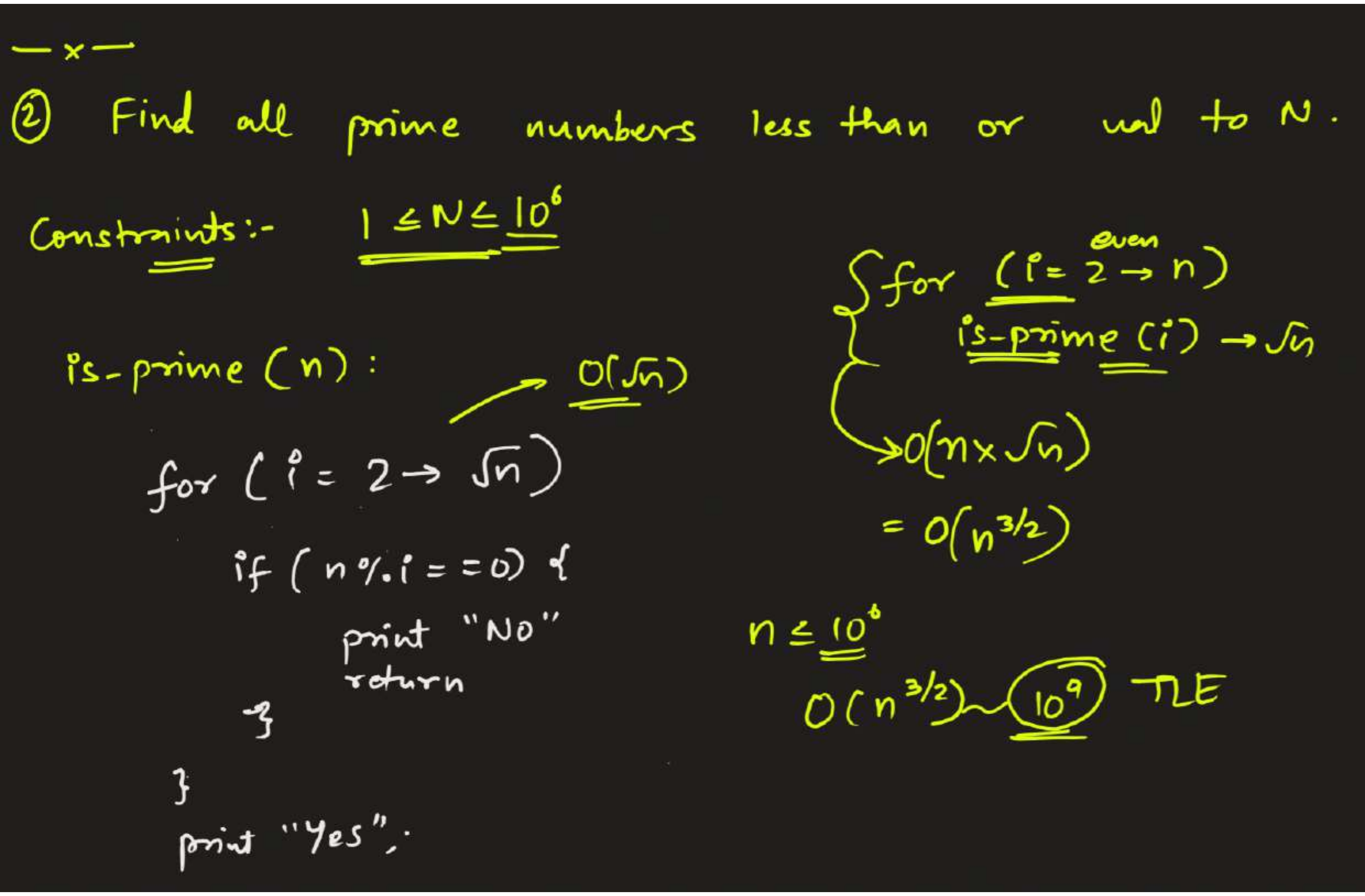
```cpp
#include<bits/stdc++.h>
using namespace std;
#define endl '\n'
#define FOR(i,a,b) for(int i=(a); i<(b); i++)
#define FORk(i,a,b,k) for(int i=(a); i<(b); i+=k)
#define RFOR(i,a,b) for(int i=(a); i>=(b); i--)
#define RFORk(i,a,b,k) for(int i=(a); i>=(b); i-=k)
void solve() {
    int n;
    cin >> n;
    if(n==1) {
        cout << "no" << endl;
        return;
    }
    if (n==2) {
        cout << "yes" << endl;
        return;
    }
    for(int i=2; i*i<=n; i++) {
        if(n%i==0) {
            cout << "no" << endl;
            return;
        }
    }
    cout << "yes" << endl;
}
```



```
int main() {
    ios_base::sync_with_stdio(false);
    cin.tie(NULL);
    cout.tie(NULL);
    int t = 1;
    cin >> t;
    while(t--) {
        solve();
    }
    return 0;
}
```

```
def solve():
    n = int(input())
    if n == 1:
        print('no')
        return
    if n == 2:
        print('yes')
        return
    i = 2
    while i*i <= n:
        if n%i == 0:
            print('no')
            return
        i += 1
    print('yes')

t = 1
t = int(input())
for i in range(t):
    solve()
```



—x—

② Find all prime numbers less than or ual to N.

Constraints:- $\underline{1 \le N \le 10^6}$

```
is-prime (n) :                    → O(√n)

    for ( i = 2 → √n )
        if ( n % i == 0) {
            print "NO"
            return
        }
    }
    print "Yes";
```

$\left\{ \text{for } (i = \overset{\text{even}}{2} \to n) \right.$

$\quad \text{is-prime}(i) \to \sqrt{n}$

$\to O(n \times \sqrt{n})$

$= O(n^{3/2})$

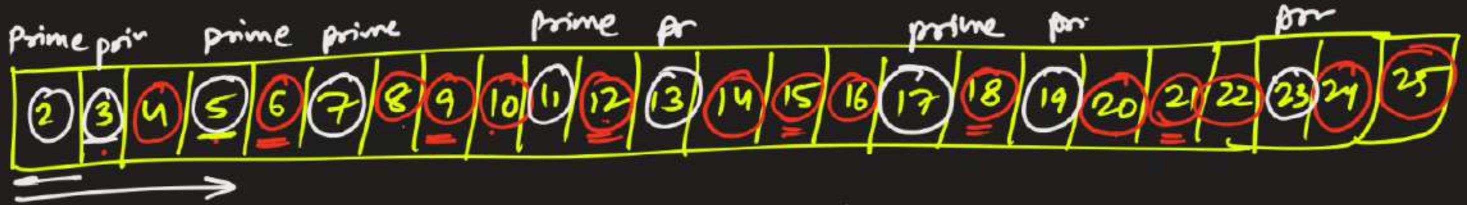
$n \le 10^6$

$O(n^{3/2}) \sim \boxed{10^9}$ TLE

Sieve Method

* skip even numbers
↳ 2 is only even prime numbers

Sieve of Eratosthenes   i=7



All the multiples of 2 are non-prime (mark them)

→ The next non-marked number is always prime

→ Mark all multiples of 3   1×11

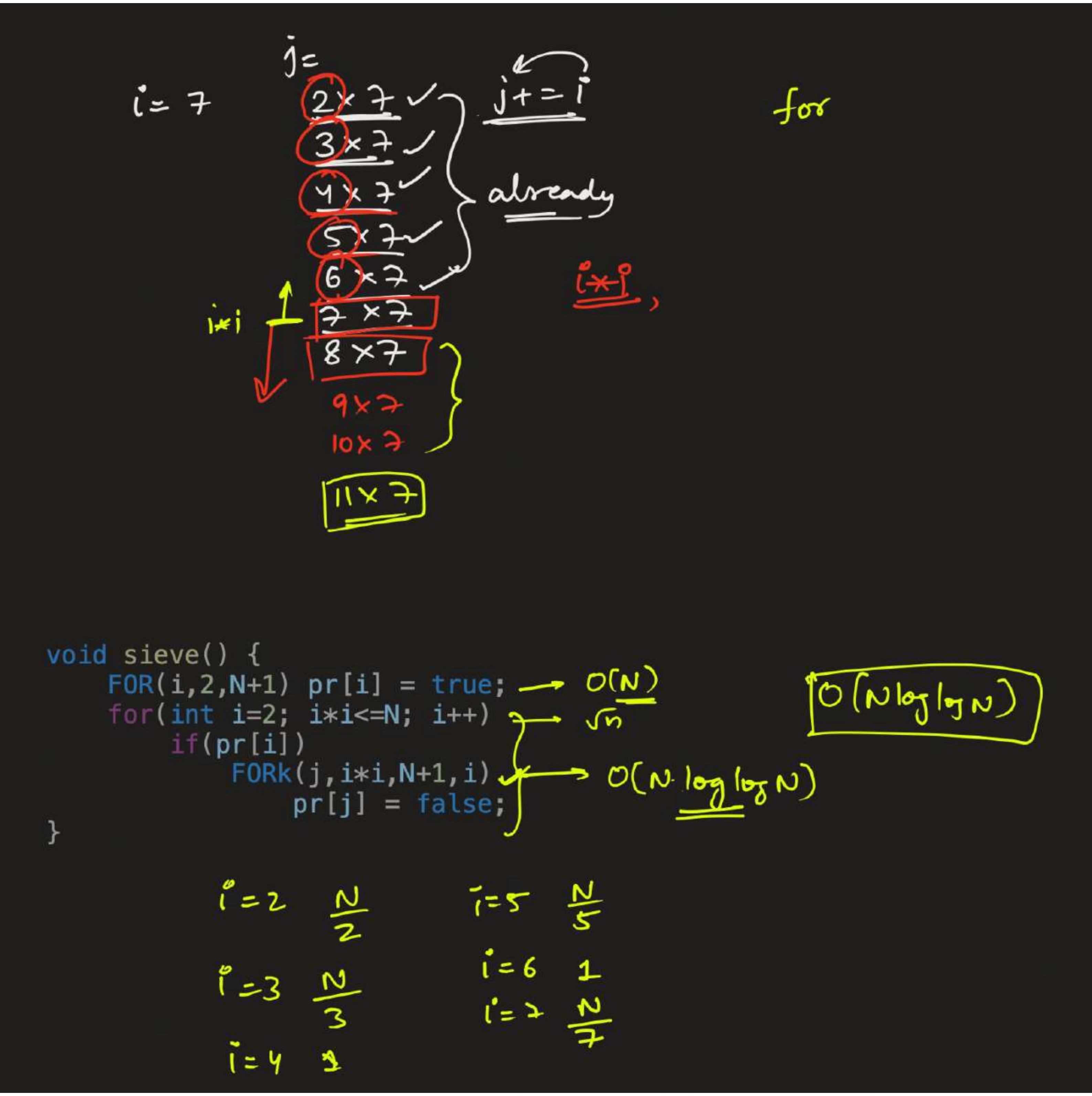
bool — true
bool — false

[True, False, ...]

```
N = 10^6
bool pr [N+1]
pr[0] = False
pr[1] = False
for (i : 2 → √n) {
    if (pr[i] == True){
        for ( j = i*i; j ≤ n ; j+= i) {
            pr [j] = False;

    }
}
```

n=25 i=5 i=7 i=
i=6 2×i
3×i



```
void sieve() {
    FOR(i,2,N+1) pr[i] = true;
    for(int i=2; i*i<=N; i++)
        if(pr[i])
            FORk(j,i*i,N+1,i)
                pr[j] = false;
}
```

$\rightarrow O(N)$

$\rightarrow \sqrt{n}$

$\rightarrow O(N \cdot \log\log N)$

$\boxed{O(N\log\log N)}$

$i=2 \quad \frac{N}{2}$

$i=3 \quad \frac{N}{3}$

$i=4 \quad 1$

$i=5 \quad \frac{N}{5}$

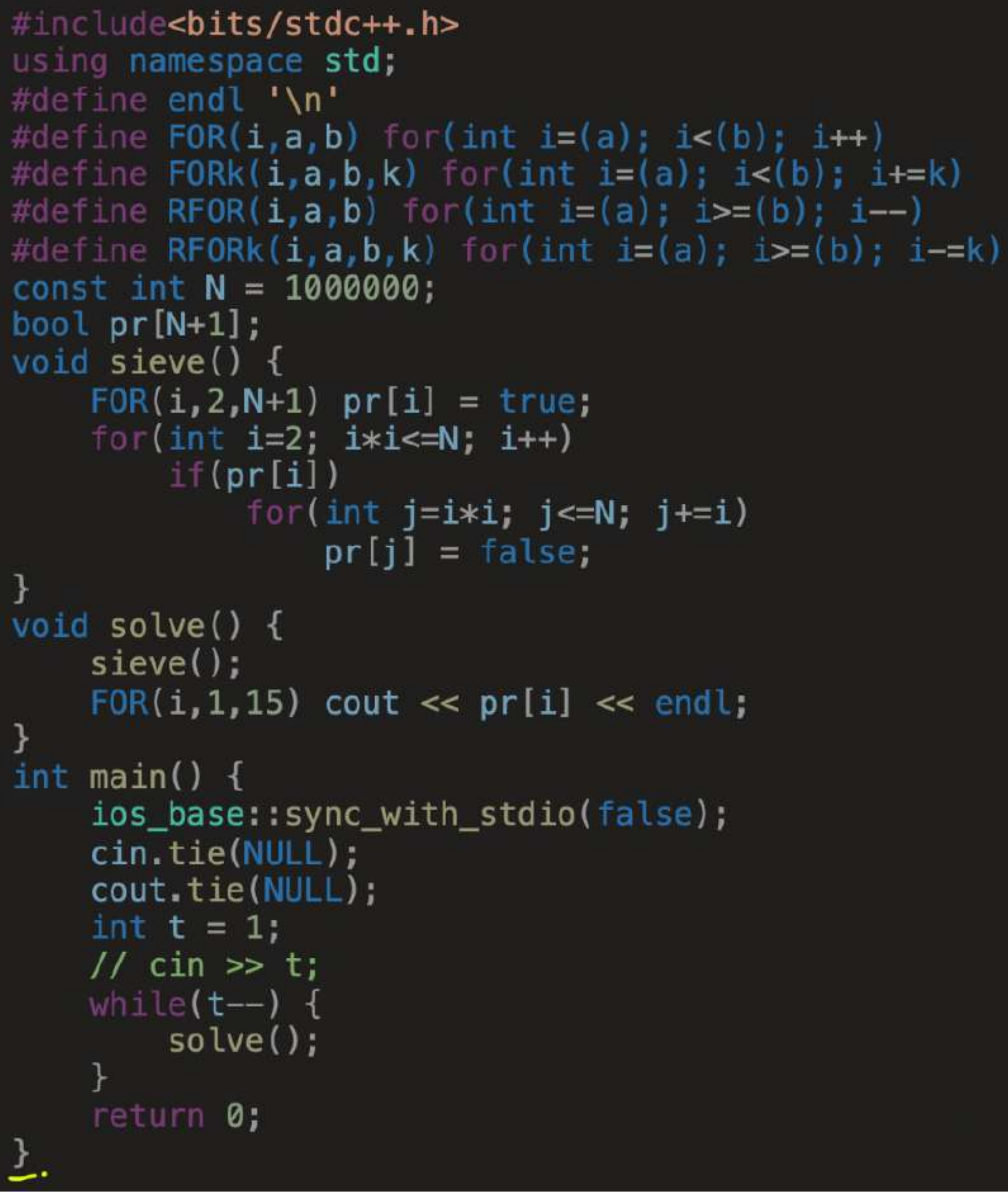
$i=6 \quad 1$

$i=7 \quad \frac{N}{7}$

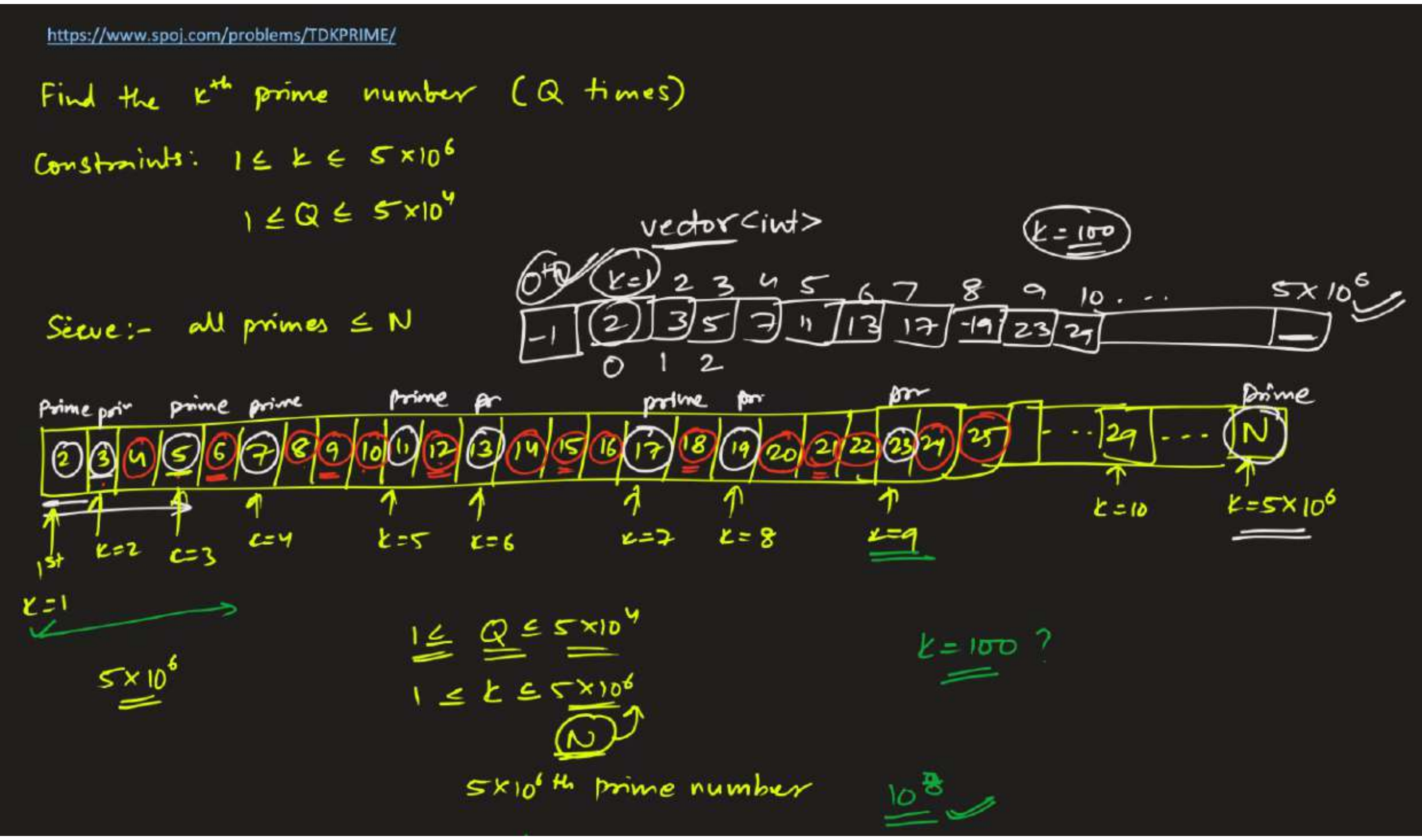
$$\left(\frac{N}{2}+\frac{N}{3}+\frac{N}{5}+\frac{N}{7}+\dots+\frac{N}{P}\right) = N\left(\underbrace{\frac{1}{2}+\frac{1}{3}+\frac{1}{5}+\dots+\frac{1}{P}}_{O(\log\log N)}\right)$$

$$= O(\underline{\underline{N\cdot\log\log N}})$$

$O(N\cdot\sqrt{n})$. $O(N\cdot\log\log N)$



```
#include<bits/stdc++.h>
using namespace std;
#define endl '\n'
#define FOR(i,a,b) for(int i=(a); i<(b); i++)
#define FORk(i,a,b,k) for(int i=(a); i<(b); i+=k)
#define RFOR(i,a,b) for(int i=(a); i>=(b); i--)
#define RFORk(i,a,b,k) for(int i=(a); i>=(b); i-=k)
const int N = 1000000;
bool pr[N+1];
void sieve() {
    FOR(i,2,N+1) pr[i] = true;
    for(int i=2; i*i<=N; i++)
        if(pr[i])
            for(int j=i*i; j<=N; j+=i)
                pr[j] = false;
}
void solve() {
    sieve();
    FOR(i,1,15) cout << pr[i] << endl;
}
int main() {
    ios_base::sync_with_stdio(false);
    cin.tie(NULL);
    cout.tie(NULL);
    int t = 1;
    // cin >> t;
    while(t--) {
        solve();
    }
    return 0;
}
```



https://www.spoj.com/problems/TDKPRIME/

Find the k^{th} prime number (Q times)

Constraints: $1 \le k \le 5\times10^6$

$1 \le Q \le 5\times10^4$

Sieve :- all primes ≤ N

vector<int>
k = 100
0th
k=1 2 3 4 5 6 7 8 9 10 . . . 5×10^6
-1 2 3 5 7 11 13 17 19 23 29
0 1 2
Prime pri prime prime prime pr prime pr pr Prime
2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 . . . 29 . . . N
1st k=1
k=2
k=3
k=4
k=5
k=6
k=7
k=8
k=9
k=10
k=5×10^6

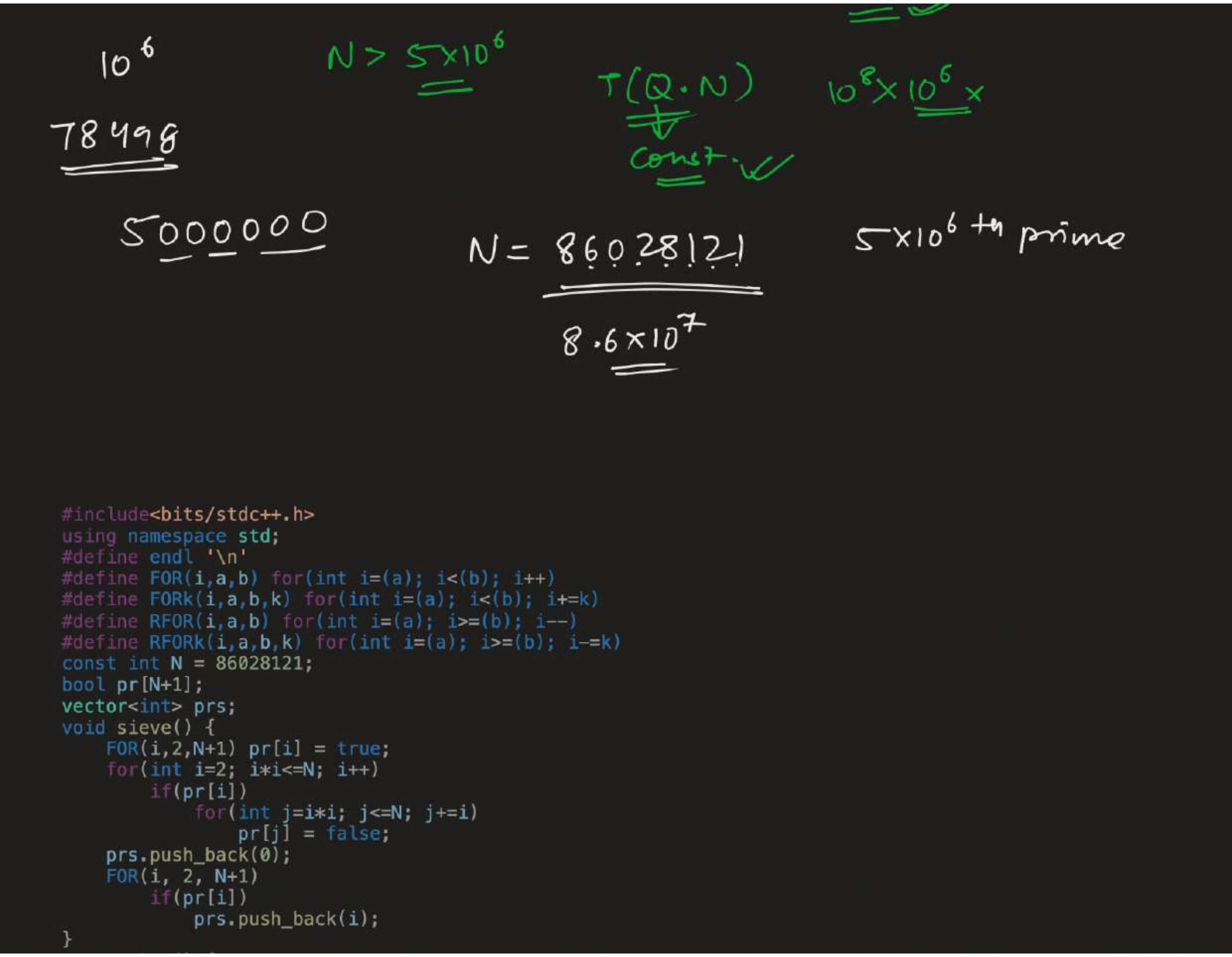
5×10^6

$1 \le Q \le 5\times10^4$

$1 \le k \le 5\times10^6$ N

k = 100 ?

5×10^6 th prime number 10^8 ✓



10^6

78498

5000000

$N > 5\times10^6$

$T(Q\cdot N)$

$\downarrow$

const. ✓

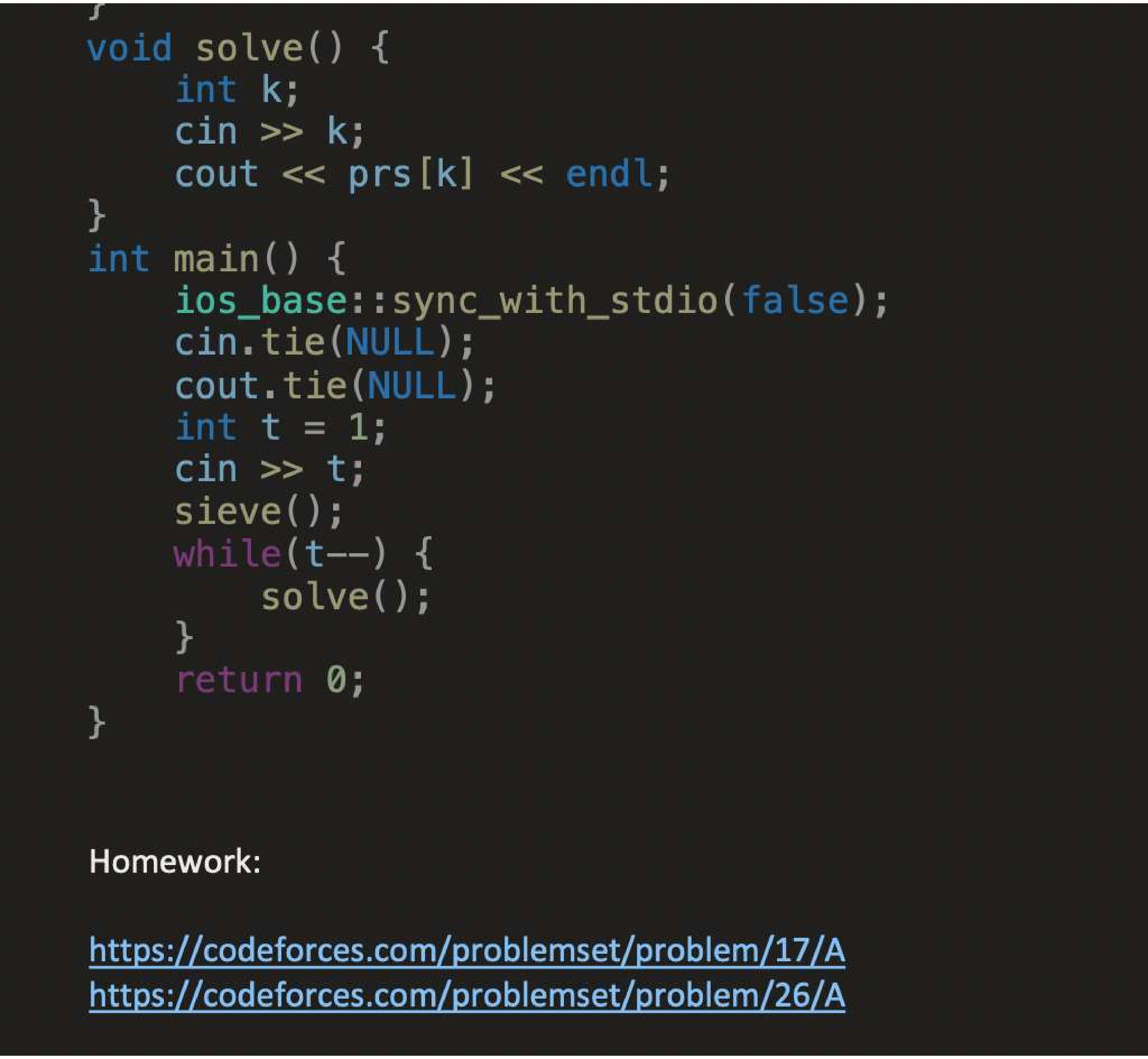
$10^8 \times 10^6$ ×

$N = 86028121$

8.6×10^7

5×10^6 th prime

```cpp
#include<bits/stdc++.h>
using namespace std;
#define endl '\n'
#define FOR(i,a,b) for(int i=(a); i<(b); i++)
#define FORk(i,a,b,k) for(int i=(a); i<(b); i+=k)
#define RFOR(i,a,b) for(int i=(a); i>=(b); i--)
#define RFORk(i,a,b,k) for(int i=(a); i>=(b); i-=k)
const int N = 86028121;
bool pr[N+1];
vector<int> prs;
void sieve() {
    FOR(i,2,N+1) pr[i] = true;
    for(int i=2; i*i<=N; i++)
        if(pr[i])
            for(int j=i*i; j<=N; j+=i)
                pr[j] = false;
    prs.push_back(0);
    FOR(i, 2, N+1)
        if(pr[i])
            prs.push_back(i);
}
```



```
}
void solve() {
    int k;
    cin >> k;
    cout << prs[k] << endl;
}
int main() {
    ios_base::sync_with_stdio(false);
    cin.tie(NULL);
    cout.tie(NULL);
    int t = 1;
    cin >> t;
    sieve();
    while(t--) {
        solve();
    }
    return 0;
}
```

Homework:

https://codeforces.com/problemset/problem/17/A
https://codeforces.com/problemset/problem/26/A