

Prime Numbers

↳ 2, 3, 5, 7, 11, ...

1 - Neither prime NOR composite

① checking Primality

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

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

$O(n)$

↓

TLE

$O(\sqrt{n})$ ✓

```
for (  $i = 2 \rightarrow n-1$  ) {
    if (  $n \% i == 0$  ) {
        print "No";
        return
    }
}
print "Yes";
```

}

$O(N)$

- x -

18

$$\begin{aligned} &= 1 \times 18 \\ 18 &= 2 \times 9 \\ &= 3 \times 6 \\ &= 6 \times 3 \\ &= 9 \times 2 \\ &= 18 \times 1 \end{aligned}$$

$$\frac{18}{2}$$

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

$$\left(\frac{n}{k}\right) \times k = n$$

$$> \sqrt{n} \quad < \sqrt{n}$$

$$\begin{aligned} 36 &= 1 \times 36 \\ &= 2 \times 18 \\ &= 3 \times 12 \\ &= 4 \times 9 \\ &= 6 \times 6 \\ &= 9 \times 4 \\ &= 12 \times 3 \\ &= 18 \times 2 \\ &= 36 \times 1 \end{aligned}$$

til \sqrt{n}

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

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

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

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

$$= 36 \times 1$$

$$O(\log n + \sqrt{n}) = \frac{O(\sqrt{n})}{\text{bits/word} \cdot \text{math.h}} \rightarrow \text{math.h}$$

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

$$n = 36$$

C++ ✓

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

Time complexity
= ?
 $O(\sqrt{n} \log n)$
 $\text{st}(n)$
↳ $\log n$

$i = 2 \rightarrow \textcircled{3} \rightarrow \sqrt{\log n}$
✓ $i \leq \sqrt{n}$
✓ $4 \rightarrow \log n$ ✓
✓ $5 \rightarrow \log n$ ✓
✓ $6 \rightarrow \log n$ ✓

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

Codechef Problem (Primality Test)

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

$$T \rightarrow 1 \leq T \leq 20$$

$$N \rightarrow 1 \leq N \leq 10^5$$

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


```
#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--> 0) {  
        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 equal to N .

Constraints:- $1 \leq N \leq 10^6$

is-prime(n):

for ($i = 2 \rightarrow \sqrt{n}$)

if ($n \% i == 0$) {
 print "No"
 return
}

}

print "Yes".

$\rightarrow \underline{O(\sqrt{n})}$

{ for ($i = 2 \rightarrow n$)
 is-prime(i) $\rightarrow \sqrt{n}$
 }
 $\rightarrow O(n \times \sqrt{n})$
 $= O(n^{3/2})$

$n \leq \underline{10^6}$

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

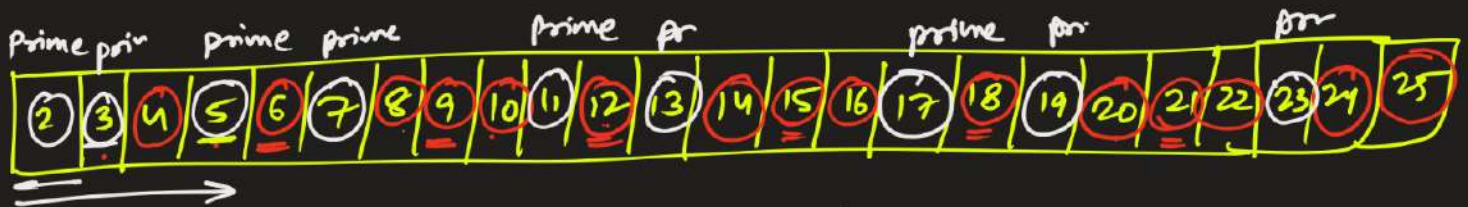
Sieve Method

Sieve of Eratosthenes

i=7

* skip even numbers

↳ 2 is only even prime numbers



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

→ The next non-marked number is always prime

→ Mark all multiples of 3

1x11

bool ← true
false

$N = 10^6$

bool pr [N+1]

[True, False, ...]

pr[0] = False

pr[i] = False

for (i : $2 \rightarrow \sqrt{n}$) {

if (pr[i] == True) {

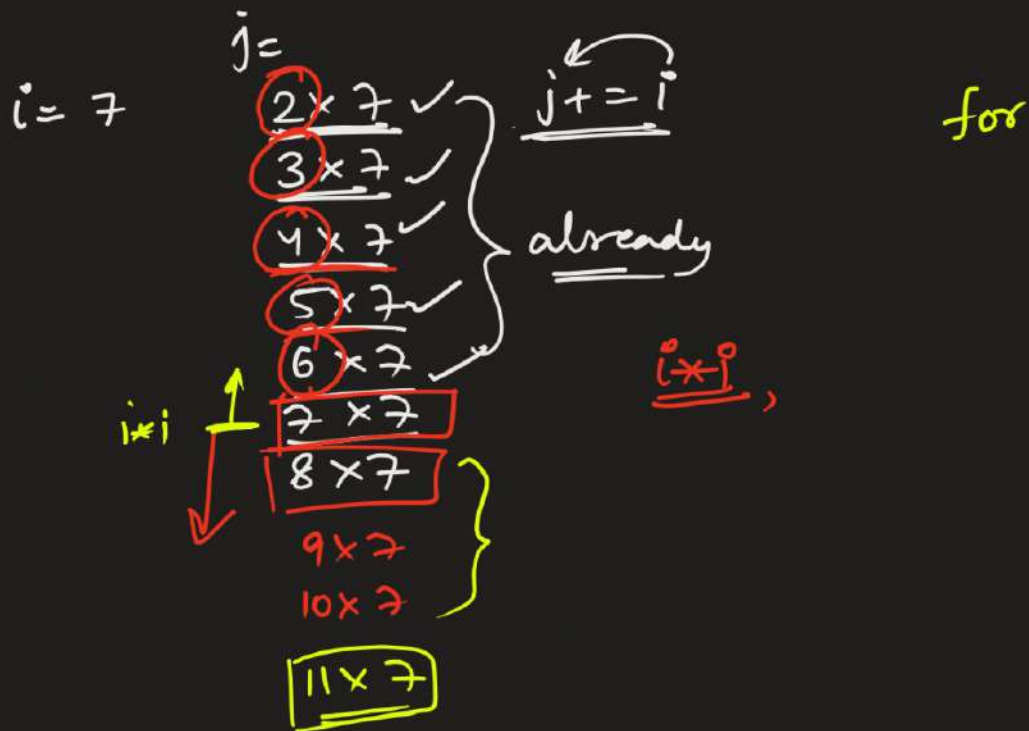
for ($j = i \times i$; $j \leq n$; $j += i$) {

pr[j] = False;

}

}

n=25 i=5 i=7 i=
i=6 $2 \times i$
 $3 \times i$



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

$O(N)$

\sqrt{n}

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

$O(N \log \log N)$

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

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

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

$$i=6 \quad 1$$

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

$$i=4 \quad 1$$

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

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

$$O(N \cdot \sqrt{N})$$

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

```
#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;
}
```

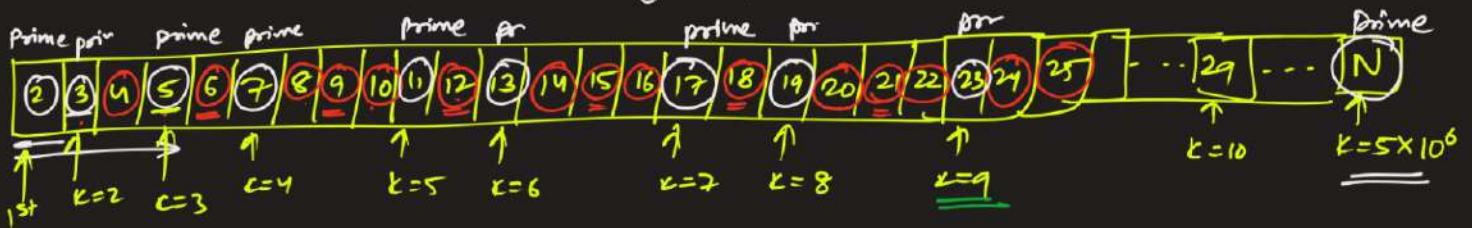
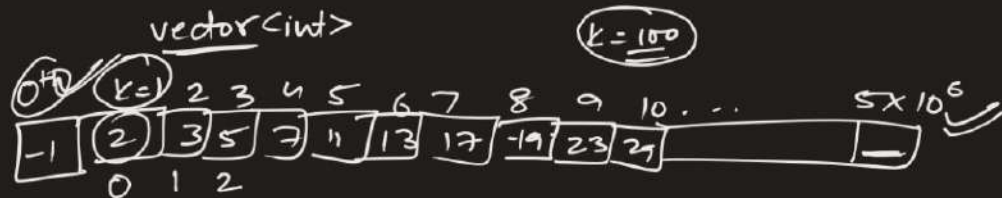
<https://www.spoj.com/problems/TDKPRIME/>

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

Constraints: $1 \leq k \leq 5 \times 10^6$

$1 \leq Q \leq 5 \times 10^4$

Sieve:- all primes $\leq N$



$k=1$

5×10^6

$1 \leq Q \leq 5 \times 10^4$

$1 \leq k \leq 5 \times 10^6$

N

5×10^6 th prime number

$k=100$?

10^8

10^6
78498

$N > \underline{5 \times 10^6}$

$T(Q \cdot N)$
Const. ✓

$\underline{10^8 \times 10^6 \times}$

5000000

$N = \underline{86028121}$
 8.6×10^7

5×10^6 + prime

```
#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>