Primary Test of O(sqrt(N))

* All divisors of a Number N occur in pairs of (a, b) s.t $a*b = N$

  Ex: 12 divisors are 1, 2, 3, 4, 6, 12
  pairs are (1, 12), (2, 6), (3, 4)

* For a divisor pair (a,b), one of them lies below sqrt(N) and other lies above sqrt(N)

(i) "Both a and b are ~~not~~ below sqrt(N)" – false because it contradicts $a*b = N$. But the statement states $a*b < N$

(ii) "Both a and b are above sqrt(N)" – false because it shows $a*b > N$ which contradicts $a*b = N$

(iii) One is below sqrt(N) and above sqrt(N) because if $b <$ sqrt(N) then $1 < sqrt(N)/b$, where $a = sqrt(N) *$ $(1+\lambda)$. Hence $a > sqrt(N)$. If $b > sqrt(N)$ then $1 > sqrt(N)/b$. Hence $a < sqrt(N)$

fact: (i) if we find of a pair, then we don't have to find anothe.
(ii) One of them lies above or below sqrt(N).

So, Time complexity is O(sqrt(N))

and main code is

```
for(int i=2; i*i<=n; i++)
```