```
#include<bits/stdc++.h>
using namespace std;

void SieveOfEratosthenes (int n)
{   bool prime [n+1];
    memset (prime, true, sizeof(prime));
    for(int p=2 ; p*p <=n; p++)
      { if (prime[p] == true)
          {
              for(int i= p*2 ; i<=n ; i=i+p)
                  prime[i] = false;
          }
      }
    for (int p = 2; p<=n ; p++)
        if(prime [p]) cout<< p<<" ";
}
int main()
{   int n =30;
    SieveofEratosthenes (n);
    return 0;
}
```

DIVISOR COUNT

$O(\sqrt{n})$

```
int divisorCount(int n)
{ int divisor = 0;
for(int i=1; i<=n; i++)
 {
  if(i*i==n) divisor++;
  else if (n%i==0)
      divisor=divisor+2;
 }
return divisor;
}
```

18 এর ডিভিজর
1, 2, 3, 6, 9, 18
√18 এর ছোট ডিভিজর
1, 2, 3 ∴ 3
Total divisor 3×2 = 6 টা

আবার কোন সংখ্যার √ যদি divisor এর সমান হয় তাহলে তাহলে 1 বাড়বে।
যেমন √16 = 4
1, 2, 3, 4 so divisor হবে 3×2 + 1 = 5টি

PRIME FACTORIZATION

$18 = 2^1 \times 3^2$ so divisors $2 \times 3 = 6$

$36 = 2^2 \times 3^2$ so divisor $(2+1) \times (2+1) = 9$

CODE

```
#include <bits/stdc++.h>
using namespace std;

vector<int> primes;

int countDivison (n)
{ int z = primes.size()/2;
  int divison = 1;
  for(int i =0; i<z; i++)
  { if (n % primes [i] == 0)
      { int cnt=1;
        while (n % prime [i] == 0)
        { n = n / prime[i];
          cnt++;
        }
        divison *= cnt;
      }
  }
  return divison;

}
```

```
int main()
{ cout<<"Enter the number to find its
                divison"<<endl;

  size_t n;
  cin>>n;
  vector<bool> sieve(n,true);
  int ub= sqrt(n)+1.5;
  for(int i=2; i<=ub; ++i)
    { if(sieve[i]==true)
      { for(size_t j=i*i; j<n; j=j+i)
          sieve[j] = false;
      }
    }
for( size_t i=2; i<=n; ++i)
   if (sieve[i]==true) primes.push_back
cout<<"Divison :"<< countDivison(n)<<en
return 0;
}
```