

```

void get_primes(){
    //外层从2~n迭代，因为这毕竟算的是1~n中质数的个数，而不是某个数是不是质数的判定
    for(int i=2;i<=n;i++){
        if(!st[i]) {
            primes[cnt++]=i;
            st[i]=i;//质数最小质因子是自己
        }
        for(int j=0;primes[j]<=n/i;j++){//primes[j]<=n/i:变形一下得到--
            primes[j]*i<=n,把大于n的合数都筛了就
            //没啥意义了
            st[primes[j]*i]=primes[j];//用最小质因子去筛合数，存最小的质数

            //1)当i%primes[j]!=0时，说明此时遍历到的primes[j]不是i的质因子，那么只
            可能是此时的primes[j]<i的
            //最小质因子，所以primes[j]*i的最小质因子就是primes[j];
            //2)当有i%primes[j]==0时，说明i的最小质因子是primes[j]，
            //因此primes[j]*i的最小质因子也就应该是prime[j]
            //之后接着用st[primes[j+1]*i]=true去筛合数时，就不是用最小质因子去更新
            了
            //因为i有最小质因子primes[j]<primes[j+1]
            //此时的primes[j+1]不是primes[j+1]*i的最小质因子
            //此时就应该退出循环，避免之后重复进行筛选。
            if(i%primes[j]==0) break;
        }
    }
}

```