

计数质数

统计所有小于非负整数 n 的质数的数量。

示例 1:

输入: $n = 10$

输出: 4

解释: 小于 10 的质数一共有 4 个, 它们是 2, 3, 5, 7。

示例 2:

输入: $n = 0$

输出: 0

示例 3:

输入: $n = 1$

输出: 0

题目简单, 但是暴力计数法会超过时间限制, 需要采用筛选法求解, 从 2 开始到 \sqrt{n} 的倍数筛掉

为啥到 \sqrt{n} 呢?

小的哪个的一个因子不会大于 \sqrt{n} , 比如说 4 的因子是 2 和 2 小于等于 2 10 的

因子 2 乘 5 最大到 3 小于 3.16227.....



```
/*
```

```
(1) 暴力法
```

```
int countPrimes(int n){
```

```

    if(n==0||n==1)
        return 0;
    int cout=0;
    for(int i=2;i<n;i++)
    {
        int j;
        for(j=2;j<i;j++)
        {
            if(i%j==0)
            {
                break;
            }
        }
        if(j==i)
        {
            cout++;
        }
    }
    return cout;
}*/

```

(2) 筛选法

```

int countPrimes(int n)
{
    if(n==0||n==1)
        return 0;
    //开辟数组空间 存放符合条件的标志位 1 不符合为 0
    int* nums=(int*)malloc(sizeof(int)*n);
    //初始化数组 默认为 1, 后期筛选
    for(int i=0;i<n;i++)
    {
        nums[i]=1;
    }
    for(int i=2;i<=sqrt(n);i++)
    {
        if(nums[i])
        {
            int k=2;
            while(k*i<n)
            {
                nums[k*i]=0;//将 2 的倍数 3 的倍数。。。sqrt(n) 的倍数置 0 排除掉之后剩余为质数
                k++;
            }
        }
    }
}

```

```
}  
//统计标志仍为 1 的数组个数  
int cout=0;  
for(int i=2;i<n;i++)  
{  
    cout+=nums[i];  
}  
return cout;  
}
```