## 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......

	2	3	4	5	6	7	8	9	10	Prime numbers			
11	12	13	14	15	16	17	18	19	20	2	3	5	7
21	22	23	24	25	26	27	28	29	30	11	13	17	
31	32	33	34	35	36	37	38	39	40				
41	42	43	44	45	46	47	48	49	50				
51	52	53	54	55	56	57	58	59	60				
61	62	63	64	65	66	67	68	69	70				
71	72	73	74	75	76	77	78	79	80				
81	82	83	84	85	86	87	88	89	90				
91	92	93	94	95	96	97	98	99	100				
101	102	103	104	105	106	107	108	109	110				
111	112	113	114	115	116	117	118	119	120				

/\*

(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++)</pre>
       int j;
       for(j=2;j<i;j++)</pre>
           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++)</pre>
    {
       nums[i]=1;
    }
    for(int i=2;i<=sqrt(n);i++)</pre>
        if(nums[i])
           int k=2;
           while(k*i<n)</pre>
               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;
}
```