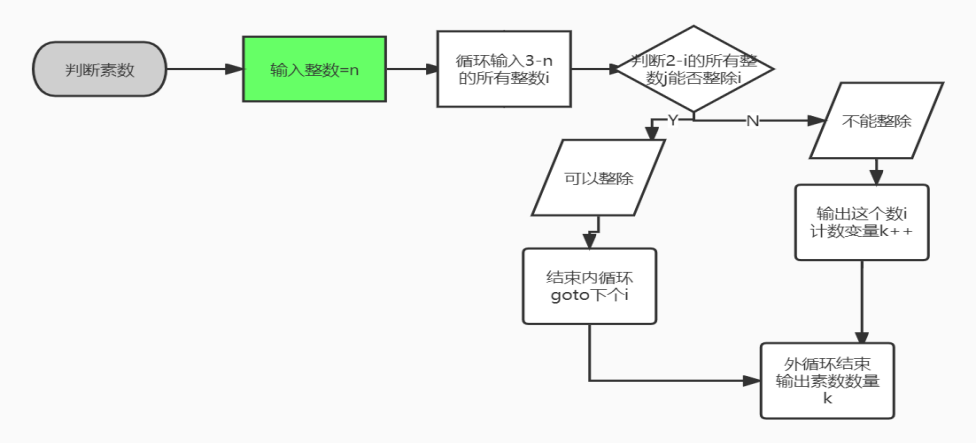
实验03

寻找n以内的所有素数以及个数

流程图



问题分析

首先这个问题是经典的循环+选择判断的问题。
要解决该问题，应该设计应该带有两个循环和一个判断的算法。
1.对此我的思路是首先接收终端输入的一个数n，然后用外循环将3-n的所有整数（i）自动输入到内循环中。
为什么是3到n？
因为2有特殊性。我们内循环的判断是从2开始依次判断能否整除。
所以，要解决2，直接在主函数前端加一句判断if（n>2）就输出2是素数，并且计数变量k自增一次即可。
2.内循环用于判断i是否是素数，从2到n的数（J）依次整除i，如果可以整除，说明i不是素数，break跳出内循环即可。如过j可以到j==i，说明i是素数，就把这个i给输出，并且计数变量k自增一次。
3.外循环结束的时候，再把计数变量k输出用于统计素数总数。

算法优化

1.判断素数的条件可以精简
由数学知识可以知道，从2开始到根号n或者n/2就可以完成判断。
2.素数会输出多次
这与输出的判断条件有关，我的方法是用一个逻辑表达式&&来保证素数只输出一次
3.可能可以用omp并行化编程来处理外循环
因为n以内各个素数的判断是相互独立的，不会一个的输出是另一个的输入，但是这样不同线程计算的i不同，所以输出的素数是乱序的。

```c
#include<stdio.h>
#include<math.h>
#include<omp.h>
int main(){
    int n,i,j,k=1,a[1000];
    printf("pelase enter a number:\n");
    scanf("%d",&n);
    printf("2 is a prime number.\n");
    #pragma omp parallel for
        for(i=2;i<=n;i++)
            {
                for(j=2;j<=i;j++)
                    {
                        if(i%j==0)break;
```