## 洛谷1579 哥德巴赫猜想升级版

### 一、题目

#### 问题描述

1742年6月7日哥德巴赫写信给当时的大数学家欧拉，正式提出了以下的猜想：任何一个大于9的奇数都可以表示成3个质数之和。质数是指除了1和本身之外没有其他约数的数，如2和11都是质数，而6不是质数，因为6除了约数1和6之外还有约数2和3。需要特别说明的是1不是质数。

这就是哥德巴赫猜想。欧拉在回信中说，他相信这个猜想是正确的，但他不能证明。

从此，这道数学难题引起了几乎所有数学家的注意。哥德巴赫猜想由此成为数学皇冠上一颗可望不可及的“明珠”。

现在请你编一个程序验证哥德巴赫猜想。

先给出一个奇数n，要求输出3个质数，这3个质数之和等于输入的奇数。

#### 输入数据

仅有一行，包含一个正奇数n，其中9<n<20000

#### 输出数据

仅有一行，输出3个质数，这3个质数之和等于输入的奇数。相邻两个质数之间用一个空格隔开，最后一个质数后面没有空格。如果表示方法不唯一，请输出第一个质数最小的方案，如果第一个质数最小的方案不唯一，请输出第一个质数最小的同时，第二个质数最小的方案。

#### 输入样例

2009

#### 输出样例

3 3 2003

#### 题目来源

洛谷 1579 https://www.luogu.com.cn/problem/P1579

### 二、题解

#### 解题思路

直接暴力枚举这三个数，三个for直接刚上去，判断三个数是否都是素数， 求和是否为目标。这样做只能得分40。

其实发现，当循环确定前两个素数时，第三个数字实际上已经可以由（n-i-j） 推出。

所以ac算法便是枚举前两个数，判断第三个数是否为正数。这样便可以ac

也可以选择把20000以内的所有素数都事先计算出来， 之后的枚举便可以只在这些素数中枚举，该算法运行效率更高。

#### 参考程序

#include <stdio.h>  
int zs (int a)  
{  
 int i;  
 if (a==1) return 0;  
 for (i=2; i\*i<=a; i++)  
 if (a%i==0) return 0;  
 return 1;  
}  
int n,a,b,c;  
int main()  
{  
 scanf("%d",&n);  
 for (a=2; a<=n-4; a++)  
 for (b=2; b<=n-4; b++)  
 {  
 c=n-a-b;  
 if (zs(a) && zs(b) && zs(c))  
 {  
 printf("%d %d %d",a,b,c);  
 return 0;  
 }  
 }  
}

#### 复杂度分析

无

#### 编程技巧

无