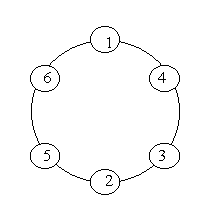
## 1016 Prime Ring Problem

### 一、题目

#### 问题描述

A ring is compose of n circles as shown in diagram. Put natural number 1, 2, …, n into each circle separately, and the sum of numbers in two adjacent circles should be a prime.

Note: the number of first circle should always be 1.

#### 输入数据

n(0<n<20).

#### 输出数据

The output format is shown as sample below. Each row represents a series of circle numbers in the ring beginning from 1 clockwisely and anticlockwisely. The order of numbers must satisfy the above requirements. Print solutions in lexicographical order.

You are to write a program that completes above process.

Print a blank line after each case.

#### 输入样例

6

8

#### 输出样例

Case 1: 1 4 3 2 5 6 1 6 5 2 3 4 Case 2: 1 2 3 8 5 6 7 4 1 2 5 8 3 4 7 6 1 4 7 6 5 8 3 2 1 6 7 4 3 8 5 2

#### 题目来源

HDU 1016 http://acm.hdu.edu.cn/showproblem.php?pid=1016

### 二、题解

#### 解题思路

按题目意思，给出一个N(0<N<20)，在1~N的所有排列中，满足相邻两个数之和是素数（头尾相邻）的排列输出。此题可用dfs即深搜做，用递归实现深搜遍历。

#### 参考程序

#include <stdio.h>  
#include <string.h>  
  
int n;  
int a[30],visit[30];  
int prime(int x) //定义一个素数环   
{  
 int flag=0;  
 if(x==1)return 0;  
 else if(x==2)return 1;  
 else   
 {  
 for(int i=2;i<x;i++)  
 {  
 if(x%i==0)  
 {  
 flag++;  
 }  
 }  
 if(flag==0)return 1;  
 else return 0;  
 }  
}  
int dfs(int num) //用递归实现深搜遍历   
{  
 if(num==n&&prime(a[0]+a[num-1])) //若满足条件则输出序列  
 {  
 for(int i=0;i<=num-1;i++)  
 {  
 if(i!=num-1)  
 printf("%d ",a[i]);  
 else printf("%d",a[i]);  
 }  
 printf("\n");  
 }  
 else //否则寻找满足的序列   
 {  
 for(int i=2;i<=n;i++)  
 {  
 if(visit[i]==0)  
 {  
 if(prime(i+a[num-1]))  
 {  
 visit[i]=1;  
 a[num++]=i; //深度优先搜索找出当前情况下的序列  
 dfs(num); //递归   
 visit[i]=0;  
 num--; //回溯   
 }  
 }  
 }  
 }  
}  
int main()  
{  
 int count=0;  
 while(~scanf("%d",&n))  
 {  
 count++;  
 printf("Case %d:\n",count);  
 a[0]=1;  
 dfs(1);  
 printf("\n");  
 }  
 return 0;  
}

#### 复杂度分析

无

#### 编程技巧

深搜