大学英语四级考试就要来临了，你是不是在紧张的复习？也许紧张得连短学期的ACM都没工夫练习了，反正我知道的Kiki和Cici都是如此。当然，作为在考场浸润了十几载的当代大学生，Kiki和Cici更懂得考前的放松，所谓“张弛有道”就是这个意思。这不，Kiki和Cici在每天晚上休息之前都要玩一会儿扑克牌以放松神经。   
“升级”？“双扣”？“红五”？还是“斗地主”？   
当然都不是！那多俗啊~   
作为计算机学院的学生，Kiki和Cici打牌的时候可没忘记专业，她们打牌的规则是这样的：   
1、  总共n张牌;   
2、  双方轮流抓牌；   
3、  每人每次抓牌的个数只能是2的幂次（即：1，2，4，8，16…）   
4、  抓完牌，胜负结果也出来了：最后抓完牌的人为胜者；   
假设Kiki和Cici都是足够聪明（其实不用假设，哪有不聪明的学生~），并且每次都是Kiki先抓牌，请问谁能赢呢？   
当然，打牌无论谁赢都问题不大，重要的是马上到来的CET-4能有好的状态。   
  
Good luck in CET-4 everybody!

Input

输入数据包含多个测试用例，每个测试用例占一行，包含一个整数n（1<=n<=1000）。

Output

如果Kiki能赢的话，请输出“Kiki”，否则请输出“Cici”，每个实例的输出占一行。

Sample Input

1

3

Sample Output

Kiki

Cici

当 n = 0 时，显然为必败点，因为此时你已经无法进行操作了

当 n = 1 时，因为你一次就可以拿完所有牌，故此时为必胜点

当 n = 2 时，也是一次就可以拿完，故此时为必胜点

当 n = 3 时，要么就是剩一张要么剩两张，无论怎么取对方都将面对必胜点，故这一点为必败点。

以此类推，最后你就可以得到；

      n    ：   0    1    2    3    4   5    6 ...

position：  P    N   N    P   N   N   P ...

你发现了什么没有，对，他们就是成有规律，使用了 P/N来分析，有没有觉得问题变简单了。

#include <iostream>

#include<stdio.h>

using namespace std;

int main()

{

int n;

while(cin>>n)

if(n%3==0)puts("Cici");else puts("Kiki");

return 0;

}

用sg函数做

#include <stdio.h>

#include <string.h>

#include<iostream>

using namespace std;

#define MAXN 1000 + 10

#define N 20

int f[N],SG[MAXN],S[MAXN];

void getSG(int n){

int i,j;

memset(SG,0,sizeof(SG));

for(i = 1; i <= n; i++){

memset(S,0,sizeof(S));

for(j = 0; f[j] <= i && j <= N; j++)

S[SG[i-f[j]]] = 1;

for(j = 0;;j++) if(!S[j]){

SG[i] = j;

break;

}

}

}

int main(){

int n;

f[0] = 1;f[1] = 2;

for(int i = 2; i <= 16; i++)

f[i] = (1<<i);

getSG(1000);

while(cin>>n)

{

if(SG[n]) printf("Kiki\n");

else printf("Cici\n");

}

return 0;

}