```c
#include <stdio.h>
#include <malloc.h>
#include <stdlib.h>
struct  Node
{
    int data;
    struct Node *next;
};
struct Node * Append (struct Node* p,int x);
struct Node * NewNode()
{
     struct Node *p;
     p = (struct Node *) malloc (sizeof (struct Node));
     if (p == NULL) {
        printf ("Error : out of memory\n");
        exit (-1);
     }
     return p;
}

int main ()
{
     int m,n,k;
     struct Node* p=NULL;//第一次的单向循环链表
     struct Node* p1=NULL;
     printf("请输入M、N、K：\n");
     scanf("%d %d %d",&m,&n,&k);
     for(int i=1;i<=m;i++){
         p=Append(p,i);
     }
     //
     struct Node* t = p->next;
     int y=n%m;//取余操作，减少循环的次数
     y--;
     int yy=y;//暂存y值
     while(p!=p->next){

         while(y--){//指针后移
             t=t->next;
             p=p->next;
         }
         y=yy;
         p1=Append(p1,t->data);
         p->next=t->next;//去掉该节点
         free(t);
         t=p->next;
     }
     p1=Append(p1,t->data);
     free(p);
     //第二次约瑟夫环开始
     struct Node* t1 = p1->next;
     int y1=k%m;//取余操作，减少循环的次数
     y1--;
     int yy1=y1;//暂存y值
     while(p1!=p1->next){
```

```
56
57          while(y1--){//指针后移
58              t1=t1->next;
59              p1=p1->next;
60          }
61          y1=yy1;
62          printf("%d ",t1->data);
63          p1->next=t1->next;//去掉该节点
64          free(t1);
65          t1=p1->next;
66      }
67          printf("%d",t1->data);
68      free(p1);
69      return 0;
70
71  }
72  struct Node* Append(struct Node* p,int x){
73      if(p==NULL){
74       struct Node *q = NewNode();
75       q->data=x;
76       q->next=q;
77       p=q;
78      }else{
79          struct Node *q = NewNode();
80          q->data=x;
81          q->next=p->next;
82          p->next=q;
83          p=q;
84      }
85      return p;
86  }
```

```
PS D:\csjjg\程序设计综合实践> cd "d:\csjjg\程序设计综合实践\" ; if ($?
) { gcc sixth.c -o sixth } ; if ($?) { .\sixth }
请输入M、N、K:
10 3 5
7 4 1 6 10 5 3 2 8 9
```