

```

1  #include <stdio.h>
2  #include <malloc.h>
3  #include <stdlib.h>
4  struct Node
5  {
6      int data;
7      struct Node *next;
8  };
9  struct Node* circle(struct Node* p,int n,int m);
10 struct Node * Append (struct Node* p,int x);
11 struct Node * NewNode()
12 {
13     struct Node *p;
14     p = (struct Node *) malloc (sizeof (struct Node));
15     if (p == NULL) {
16         printf ("Error : out of memory\n");
17         exit (-1);
18     }
19     return p;
20 }
21
22 int main ()
23 {
24     int m,n[10000],i=0;
25     struct Node* p=NULL;//第一次的单向循环链表
26     printf("请输入M和报数密码，报数密码以0结束：\n");
27     scanf("%d",&m);
28     scanf("%d",&n[0]);
29     while(n[i]!=0){
30         i++;
31         scanf("%d",&n[i]);
32     }//输入密码
33     for(int j=1;j<=m;j++){
34         p=Append(p,j);
35     }
36     //
37     for(int j=0;j<=i-2;j++){
38         p=circle(p,n[j],m);
39     }
40     //第二次约瑟夫环开始
41     struct Node* p1=p;
42     struct Node* t1 = p1->next;
43     int y1=n[i-1]%m;//取余操作，减少循环的次数
44     y1--;
45     int yy1=y1;//暂存y值
46     while(p1!=p1->next){
47
48         while(y1--){//指针后移
49             t1=t1->next;
50             p1=p1->next;
51         }
52         y1=yy1;
53         printf("%d ",t1->data);
54         p1->next=t1->next;//去掉该节点
55         free(t1);

```

```

56         t1=p1->next;
57     }
58     printf("%d",t1->data);
59     free(p1);
60     return 0;
61
62 }
63 struct Node* circle(struct Node* p,int n,int m){//除最后一个密码之外，前几个密码
    的约瑟夫环操作在此执行
64     struct Node* p1=NULL;
65     struct Node* t = p->next;
66     int y=n%m;//取余操作，减少循环的次数
67     y--;
68     int yy=y;//暂存y值
69     while(p!=p->next){
70
71         while(y--){//指针后移
72             t=t->next;
73             p=p->next;
74         }
75         y=yy;
76         p1=Append(p1,t->data);
77         p->next=t->next;//去掉该节点
78         free(t);
79         t=p->next;
80     }
81     p1=Append(p1,t->data);
82     free(p);
83     return p1;
84 }
85 struct Node* Append(struct Node* p,int x){
86     if(p==NULL){
87         struct Node *q = NewNode();
88         q->data=x;
89         q->next=q;
90         p=q;
91     }else{
92         struct Node *q = NewNode();
93         q->data=x;
94         q->next=p->next;
95         p->next=q;
96         p=q;
97     }
98     return p;
99 }

```

```

PS D:\csjjg> cd "d:\csjjg\程序设计综合实践\" ; if ($?) { gcc seventh.c
-o seventh } ; if ($?) { .\seventh }
请输入M和报数密码，报数密码以0结束：
10 3 5 2 0
4 6 5 2 9 1 3 7 8 10
PS D:\csjjg\程序设计综合实践>

```