*1、100个小朋友围成一个圈，设定编号为1~100，依次按1、2、3、4、5、6、7、8、9循环报数，报到9的出圈，请写代码打印出各个小朋友出圈顺序。*

**答：**

length = int(input('please input the list length:'))

l=[]

for i in range(1,length+1):

l.append(i)

def remove(lst):

l\_remove=[]

for x in range(8,len(lst),9):

print(lst[x],end='-->')

l\_remove.append(lst[x])

for x in l\_remove:

lst.remove(x)

if len(lst)>=9:

remove(lst)

remove(l)

2、*如下图所示，有一片棋盘格子样的街区，每个十字路口可能有一个传送门结束游览，用0表示，或者可能有一些金币供拾取，用地图上的数字表示。从左上角出发，只能向下或者向右移动，求给定地图是否存在某个终点为传送门的路径可以获得指定数额的金币N。*

\*---5-----1---4----15---23—4---8---1---15

| | | | | | | | | |

1---0-----3---12---1-----0---9---3---4----8

| | | | | | | | | |

4---5-----7----0---58----5---0---7---9-----3

| | | | | | | | | |

7---11-----5---1----4---72---1---2---9---3

**答：**以下是用C++实现代码：

基本思路是利用递归算法实现

\*/

#include <iostream>

using namespace std;

//二维数组array中找number

void seek(int array[4][10],int number);

//向右找，array仍然是原来的数组，a，b表示目前所在的左上角坐标，number表示要找的数

void seekRight(int array[4][10],int a,int b,int number);

//向左找，array仍然是原来的数组，a，b表示目前所在的左上角坐标，number表示要找的数

void seekDown(int array[4][10],int a,int b,int number);

//判断是否找到的标志，初始化为false

bool finded = false;

int main()

{

//初始化数组

int array[4][10] =

{

{0,5,1,4,15,23,4,8,1,15},

{1,0,3,12,1,0,9,3,4,8},

{4,5,7,0,58,5,0,7,9,3},

{7,11,5,1,4,72,1,2,9,3}

};

//用于存储输入的数，先初始化为0

int number = 0;

do

{

finded = false;

//输入数

cout<<"please input a number:";

cin>>number;

cout<<"输入的是："<<number<<endl;

//找数

seek(array,number);

//如果找到

if(finded)

cout<<"finded!!!"<<endl;

//否则

else

cout<<"not finded!!!"<<endl;

}while(number != -1);

}

void seek(int array[4][10],int number)

{

//一开始左上角坐标是数组array[0][0];所以a = 0，b = 0；

int a = 0,b = 0;

//向右找这个数

seekRight(array,a,b,number);

//向左找数

seekDown(array,a,b,number);

}

void seekRight(int array[4][10],int a,int b,int number)

{

//如果找到了

if(number == array[a][b+1])

{

finded = true;

}

else if(number>array[a][b+1])

{

//递归调用，不过数组的左上角坐标得发生变化，由于是向右找所以a不变，b+1，同时变成找number-array[a][b+1]，其他类似

seekRight(array,a,b+1,number-array[a][b+1]);

seekDown(array,a,b+1,number-array[a][b+1]);

}

else if(number < array[a][b+1])

{

//do nothing

}

}

void seekDown(int array[4][10],int a,int b,int number)

{

if(number == array[a+1][b])

{

finded = true;

}

else if(number>array[a+1][b])

{

seekRight(array,a+1,b,number-array[a+1][b]);

seekDown(array,a+1,b,number-array[a+1][b]);

}

else if(number<array[a+1][b])

{

// do nothing

}

}