# 实习报告

题目:车厢调度

班级：电信1809 姓名：马奇逸 学号： U201813500 日期：2019.4.8

#### 需求分析

1.假设停在铁路调度站入口处的车厢序列的编号依次为1，2，3，4，···，n。设计一个程序，求出所有可能由此输出的长度为n的车厢序列。

（1）输入整形变量，范围1~4。

（2）将站内元素按整形变量输出。

（3）输出所有可能由此输出的长度为n的车厢序列。

（4）数据测试：1，2，3，4。

#### 概要设计

1. 定义栈

struct snode

{

int data[MaxLen];

int bottom;

}s;//定义栈

1. 主程序

void main()

{

int path[MaxLen];

printf("输入车厢总数：\n");

scanf("%d",&n);

Initstact();

push(1);

printf("所有序列：\n");

process(1,path,0);//从1开始，将元素放进每个栈底，递归

}

1. 主程序main调用 Initstact( ); push( ); process( ， ， );

process( ， ， ) 函数调用push( )；pop( )；Emptys( )；

#### 三．详情设计

1. 定义栈

struct snode

{

int data[MaxLen];

int bottom;

}s;

2.开栈

void Initstact()

{

s.bottom=-1;

}

3.进栈

void push(int q)

{

s.bottom++;

s.data[s.bottom]=q;

}

4.出栈

int pop()

{

int temp;

temp=s.data[s.bottom];

s.bottom--;

return temp;

}

5.判断是否调度铁轨为空

int Emptys()

{

if(s.bottom==-1)

return 1;

else

return 0;

}

6.二重递归

void process(int pos,int path[],int curp)

{

int m,i;

if(pos<n){

push(pos+1);

process(pos+1,path,curp);

pop();

}

if(!Emptys()){

m=pop();

path[curp]=m;

curp++;

process(pos,path,curp);

push(m);

}

if(pos==n&&Emptys()){

for(i=0;i<curp;i++){

printf("%d,",path[i]);

}

printf("\n");

}

}

7.主函数

void main()

{

int path[MaxLen];

printf("输入车厢总数：\n");

scanf("%d",&n);

Initstact();

push(1);

printf("所有序列：\n");

process(1,path,0);

}

#### 四．调试分析

1.递归逻辑错误

解决方式：加断点观察入栈出栈的值。

#### 五．用户使用说明

在显示“输入车厢总数：”后输入车厢总数n的值。

#### 六．测试结果

1. 输入：1

输出：1

1. 输入：2

输出：2，1，

1，2，

1. 输入：3

输出：3,2,1,

2,3,1,

2,1,3,

1,3,2,

1,2,3,

1. 输入：4

输出：4,3,2,1,

3,4,2,1,

3,2,4,1,

3,2,1,4,

2,4,3,1,

2,3,4,1,

2,3,1,4,

2,1,4,3,

2,1,3,4,

1,4,3,2,

1,3,4,2,

1,3,2,4,

1,2,4,3,

1,2,3,4,

#### 七．附录

1. 源代码

#include<stdio.h>

#define MaxLen 100

int n;

struct snode

{

int data[MaxLen];

int bottom;

}s;//定义栈

void Initstact()//定义栈中数组与值的关系

{

s.bottom=-1;

}

void push(int q)//进栈

{

s.bottom++;

s.data[s.bottom]=q;

}

int pop()//出栈

{

int temp;

temp=s.data[s.bottom];

s.bottom--;

return temp;

}

int Emptys()

{

if(s.bottom==-1)

return 1;

else

return 0;

}

void process(int pos,int path[],int curp)//curp为出栈铁轨的位置编号

{

int m,i;

if(pos<n){//若元素未全放入，则继续入栈

push(pos+1);

process(pos+1,path,curp);//递归入栈

pop();//下一个元素处理运算完后，准备出栈

}

if(!Emptys()){//递归出栈

m=pop();

path[curp]=m;

curp++;

process(pos,path,curp);//出栈后处理下一个元素进栈

push(m);

}

if(pos==n&&Emptys()){//输出一个可能方案

for(i=0;i<curp;i++){

printf("%d,",path[i]);

}

printf("\n");

}

}

void main()

{

int path[MaxLen];

printf("输入车厢总数：\n");

scanf("%d",&n);

Initstact();

push(1);

printf("所有序列：\n");

process(1,path,0);//从1开始，将元素放进每个栈底，递归

}