**实习报告**

题目：设计一个能求出所有可能由此输出的长度为n的车厢序列

姓名：李林徽 班级：电信1808 学号：201813472 完成日期：2019.4.8

1. 需求分析

将n个数进行栈方式的全排列

1. 概要设计

定义了一个栈a和一个输出队列OUTPUT，将需要处理的元素用栈和队列进行操作。

1. 详细设计

int a[N];

int OUTPUT[N];

void solve(int i,int t,int m,int n)

{

if(n==t&&m==0)//输出

if(m>0)//出栈

if(i<=t)//入栈

}

Int main()

1. 调试分析
2. 将3,4等好几个数进行测试，都有点问题，发现原来是出栈再入栈没有还原原来的值，所以后来用了x将原来的数进行储存
3. 出栈和入栈的顺序也是有点纠结的，经过实际跑过之后才确定的，之后再经过思考发现这样的顺序还是有道理的
4. 由于多次递归，top的不太好更改，所以直接调用了一个m来计数
5. 用户使用说明
6. 本程序的执行文件为“车厢调度.c”
7. 用户可以更改N的值（在程序中直接修改）来实现该用法
8. 测试结果

Possiblity~ 123

Possiblity~ 132

Possiblity~ 213

Possiblity~ 231

Possiblity~ 321

1. 附录

//i 即将要压入的元素，第一个为1，以后序号依次递增

//t 火车总数

//m 栈内元素个数

//n "输出队列"中的个数

#include <stdio.h>

#include <stdlib.h>

int OUTPUT[10];//输出

int a[10];//栈

void solve(int i,int t,int m,int n)

{

if(n==t&&m==0)

{

printf("Possiblity~ ");

int j;

for(j=0;j<t;j++)

{

printf("%d",OUTPUT[j]);

}

printf("\n");

}

if (m>=1)//出栈

{

OUTPUT[n]=a[m-1];

solve(i,t,m-1,n+1);

}

if(i<=t)//入栈

{

int x;

x=a[m];

a[m]=i;

solve(i+1,t,m+1,n);

a[m]=x;

}

}

int main()

{

solve(1,3,0,0);

return 0;

}