题目：编制一个车厢调度问题的程序班级：电信1808班 姓名：郭佳璐 学号：U201813461 完成日期：2019年4月8日

1. 需求分析
2. 以一维数组str[n]表示n节车厢，str[i]表示第i+1节车厢，限定n大于0小于100。
3. 用户已整形的形式输入n的值。
4. 结果以整型的形式输出。
5. 本程序以输出n节车厢的所有可能由铁路调度站输出的车厢序列。
6. 测试数据：

输入：n=4；

输出：

1:1234

2:1243

3:1324

4:1342

5:1432

6:2134

7:2143

8:2314

9:2341

10:2431

11:3214

12:3241

13:3421

14:4321

1. 概要设计
2. 本程序包含三个模块
3. （1）主程序模块：

Int main()

{

初始化；

接受命令；

处理命令；

}

1. 全排列（perm）模块 实现n节车厢的全排列
2. 出栈模块 判断整数序列 str[] 是否满足进出栈规则

各模块之间的调用关系如下

主程序模块

↓

全排列模块

↓

出栈模块

1. 求解车厢排序中一种排序的伪码算法；

通过前后车厢调换顺序得到一种排列，并判断此排列中的每个元素其后比它小的数是否为降序序列，如果是则输入此排序，否则不输出。

1. 详细设计

1.全排列类型

void perm(int str[],int k,int n)

1. 出栈类型

void print(int str[],int n)

1. 全排列算法：

void perm(int str[],int k,int n)

{

int i,temp;

if(k==n-1)print(str,n);//k和n-1相等，即一趟递归走完

else{

for(i=k;i<n;i++){//把当前节点元素与后续节点元素交换

temp=str[k]; str[k]=str[i]; str[i]=temp;//交换

Perm(str,k+1,n);//把下一个节点元素与后续节点元素交换

temp=str[i]; str[i]=str[k]; str[k]=temp;//恢复原状

}

}

4.出栈算法：

void print(int str[],int n){

int i,j,k,l,m,flag=1,b[2];

for(i=0;i<n;i++) // 对每个str[i] 判断其后比它小的数是否为降序序列

{

m=0;

for(j=i+1;j<n&&flag;j++){

if (str[i]>str[j])

{

if (m==0) b[m++]=str[j];//记录str[i]后比它小的数

else

{

//如果之后出现的数比记录的数还大，改变标记变量

if (str[j]>b[0]) flag=0;

//否则记录这个更小的数

else b[0]=str[j];

}

}

}

}

if(flag) // 满足出栈规则则输出 str[] 中的序列

{

printf(" %2d:",cont++); //输出序号

for(i=0;i<n;i++)

printf("%d",str[i]);//输出序列

printf("\n");

}}

1. 主函数：

int main(){

int str[100],n,i;

scanf("%d",&n);

for(i=0;i<n;i++)

str[i]=i+1;

Perm(str,0,n);

printf("\n");

return 0;}

1. 调试分析：

将n节车厢全排列后，得到的序列需满足其中每一个元素后面比它小的元素呈递减排列，如此才满足出栈规则。

1. 用户手册：

用户只需通过键盘输入n的值，运行程序，即可得到满足题目需求的所有排列结果。

1. 调试结果：

输入：n=4；

输出：

1:1234

2:1243

3:1324

4:1342

5:1432

6:2134

7:2143

8:2314

9:2341

10:2431

11:3214

12:3241

13:3421

14:4321

七．附录