**“车厢调度”实验报告**

题目：数据结构题集p98的“车厢调度”

班级：电信1808

姓名：瞿博文

学号：U201813460

日期：2019年3月25日星期一

1. 需求分析

程序设计的任务：问题描述]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……N。设计一个程序，求出所有由此输出的长度为N的车厢序列

1. 输出的形式：通过自己定义的print函数进行输出可能的序列
2. 程序所达到的功能：可以得到所有由此输出的长度为N的车厢序列
3. 测试数据：

输入：n=1

输出：1

输入：n=2

输出：2 1

1 2

输入：n=3

输出：the number is

3 2 1

2 3 1

2 1 3

1 3 2

1 2 3

1. 概要设计

ADT：栈的顺序存储

主程序：主函数进行输入n的值，调用建立栈的函数，定义两个数组，一个数组a是进行存储序号1，2，3，...，n的，另一个进行打印结果序列。调用f函数（实现递归的函数）。

各程序模块：栈的基本操作函数：建立，入栈，出栈，判断栈是否为空，以及f函数（实现递归的函数）和进行打印结果序列的print函数

三：详细设计

#include<stdio.h>

#include<stdlib.h>

#define Maxsize 10

typedef struct

{

int \*base;

int \*top;

int stacksize;

}SqStack;

int InitStack(SqStack \*pS)

{

pS->base=(int \*)malloc(Maxsize\*sizeof(int));

pS->top=pS->base;

pS->stacksize=Maxsize;

return 1;

}

int Push(SqStack \*pS,int e)

{

\*(pS->top)++=e;

return 1;

}

int Pop(SqStack \*pS,int \*e)

{

if(pS->top==pS->base)return 0;

\*e=\*--pS->top;

return 1;

}

int StackEmpty(SqStack \*pS)

{

if(pS->top==pS->base)return 1;

else return 0;

}

void print(int b[],int n)

{

int k;

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

printf("%d\t",b[k]);

printf("\n");

}

void f(SqStack \*pS,int a[],int b[],int p,int q,int n)

{

if(p<n)

{

Push(pS,a[p]);

f(pS,a,b,p+1,q,n);

Pop(pS,&b[q]);

}

if(!StackEmpty(pS))

{

Pop(pS,&b[q]);

f(pS,a,b,p,q+1,n);

Push(pS,b[q]);

}

if(q>=n&&StackEmpty(pS))print(b,n);

}

void main()

{

int i,n;

int a[Maxsize],b[Maxsize];

printf("please input the number n = \n");

scanf("%d",&n);

SqStack S;

InitStack(&S);

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

a[i]=i+1;

printf("the number is \n");

f(&S,a,b,0,0,n);

}

四．调试分析

（1）：问题1：如何理解此题

解答：此题本质上就是1，2，3，4，...，n按顺序入栈，求所有可能的出栈序列。而对每一个数据都有入和出两种操作（除第一个和最后一个），对一次入栈操作，下一次可能是再入栈（若未满）或该入栈数据出栈；对一次出栈操作，下一次则可能是再出栈（若未空）或入栈。这就符合基本的递归特性。P.S.：对第一个数据一定是入栈，对最后一个操作一定是出栈。

问题2：栈的顺序存储以及其基本操作

解决：定义base（栈底）和top（栈顶）指针以及栈的最大长度，base==top则为空，top-base==最大长度则为满，插入和删除时直接对top指针进行操作即可。

1. 时间复杂度和空间复杂度分析

时间复杂度：O（n！） 空间复杂度：O（n）

1. 经验和体会
2. 对递归应用更熟悉
3. 对栈的基本操作很熟悉了
4. 用户使用说明

步骤一：书写栈的顺序存储的结构体定义

步骤二：书写栈的基本操作的函数板块

步骤三：书写进行递归操作（题目主要操作）的f函数

步骤四：书写打印结果序列的print函数板块

步骤五：书写主函数模块并调用所需函数

步骤六：输入n值，即可输出结果序列

1. 测试结果

输入：n=1

输出：1

输入：n=2

输出：2 1

1 2

输入：n=3

输出：the number is

3 2 1

2 3 1

2 1 3

1 3 2

1 2 3

输入：n=4

输出：the number is

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