1. 需求分析

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，求出所有可能的长度为n的车厢序列。

实现栈的顺序存储结构SqStack之上实现栈的基本操作，即实现栈类型。程序对栈的基本操作必须借助于基本操作进行。

测试数据取n=3，4，程序输出结果应该在屏幕上显示出来。

1. 概要设计

具体要求：火车调度问题可以用栈来解决，而核心就是递归。

设计思想：一个数进栈以后，要么立刻出栈要么立刻下一个数进栈，出栈以后，也有两种处理方式：要么立刻出栈。要么下一个数进栈。

核心算法：包括两重递归。

各个模块功能：

1. 详细设计

#include "pch.h"

#include<stdio.h>

#include<stdlib.h>

int final;

int total = 0;

typedef struct

{

int \*base;

int \*top;

int size;

}sqs;

int Inits(sqs \*s)

{

s->base = (int\*)malloc(final \* sizeof(int));

if (!s->base) exit(0);

s->top = s->base;

s->size = final;

}

void Push(sqs \*s, int e)

{

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

}

int Pop(sqs \*s)

{

if (s->top == s->base)

{

return 0;

}

else

{

return \*(--(s->top));

}

}

int Empty(sqs \*s)

{

if (s->top == s->base)

{

return 1;

}

else

{

return 0;

}

}

int Fulls(sqs \*s)

{

if (s->top - s->base == final)

{

return 1;

}

else

{

return 0;

}

}

void Print(sqs s)

{

int \*p;

p = s.base;

printf("\t[%d]:", total);

for (; p != s.top; )

{

printf("%d", \*p++);

}

printf("\n");

}

void search(sqs \*input, sqs \*temp, sqs \*output)

{

if (!Empty(input))

{

Push(temp, Pop(input));

search(input, temp, output);

Push(input, Pop(temp));

}

if (!Empty(temp))

{

Push(output, Pop(temp));

search(input, temp, output);

Push(temp, Pop(output));

}

if (Fulls(output))

{

total++;

Print(\*output);

}

}

void main()

{

sqs input, temp, output;

int i;

printf("请输入车厢长度:\n");

scanf\_s("%d", &final);

printf("车厢序列为:\n");

Inits(&input);

Inits(&temp);

Inits(&output);

for (i = final; i >= 1; i--)

{

Push(&input, i);

}

search(&input, &temp, &output);

}

1. 调试分析

无明显问题

1. 用户使用说明
2. 测试结果

321 231 213 123

7.附录

#include "pch.h"

#include<stdio.h>

#include<stdlib.h>