**《C语言课程设计》实验报告**

专业：­ 计算机科学与技术 班级： 1803011

姓名： 张帅豪 学号： 18030100101 分工：50%

姓名： 赵宇轩 学号： 18030100103 分工：50%

1. **问题描述**

**题目2：**简单数据结构-堆栈模拟

**涉及知识点：**内存管理、结构体定义、基本数据结构

**要求：**编写一个程序模拟堆栈，要求能够模拟、入栈、出栈、返回栈顶元素等基本操作。栈中元素可用整数代替。不能使用C++模板库预定义的类型。程序运行中可输入多组入栈、出栈操作，每次操作后展示栈中元素。

1. **技术关键点分析**

1. 初始栈函数

2. 遍历栈函数

3.进栈函数

4. 出栈函数

5.获得栈顶元素函数

6. Main函数使用

请输入选项1.进栈 2.出栈 3.遍历栈 4.获得栈顶元素 0.退出

* 初始栈函数

定义栈结构

typedef struct node   
{   
 int value; //数据域   
 struct node \*Next;//指针域   
}Node;   
// 定义一个栈结构   
typedef struct stack   
{   
 Node \*Top; //栈顶   
 Node \*Bottom; //栈底   
}Stack;   
初始栈函数

//初始化栈   
void InitStack(Stack \*s)   
{   
 s->Top = (Node\*)malloc(sizeof(Node)); // 分配内存空间给栈顶   
 s->Bottom = s->Top; // 使栈底也指向栈顶空间   
 s->Top->Next = NULL; // 栈顶指针置为NULL；   
 return ;   
}

* 遍历栈函数

//遍历栈的函数   
void PrintStack(Stack \*s)   
{   
 Node\* p1 = s->Top;   
 while(p1!= s->Bottom) // 只要栈顶不等于栈底，循环   
 {   
 printf("%d ",p1->value); // 打印栈顶的成员member   
 p1 = p1->Next; // 栈顶指针向下移动一次   
 }   
}

* 进栈函数

//进栈   
void Push(Stack \*s,int value)   
{   
 Node\* p1 = (Node\*)malloc(sizeof(Node)); // 定义一个新节点，   
 p1->value = value;   
 p1->Next = s->Top; //使新节点的指针指向栈顶   
 s->Top = p1; //把新节点作为新栈顶   
}

* 出栈函数

//出栈   
int Pop (Stack \*s){   
 int value = s->Top->value;   
 s->Top = s->Top->Next;   
 return value;   
   
}

* 获得栈顶元素函数

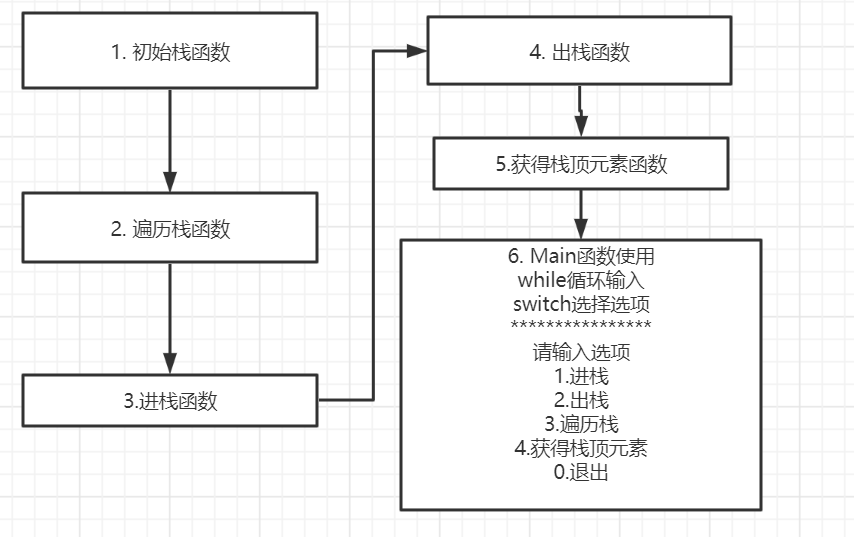
//获得栈顶元素   
int GetTop(Stack \*s){   
 return s->Top->value;   
   
}

* Main函数使用

请输入选项1.进栈 2.出栈 3.遍历栈 4.获得栈顶元素 0.退出

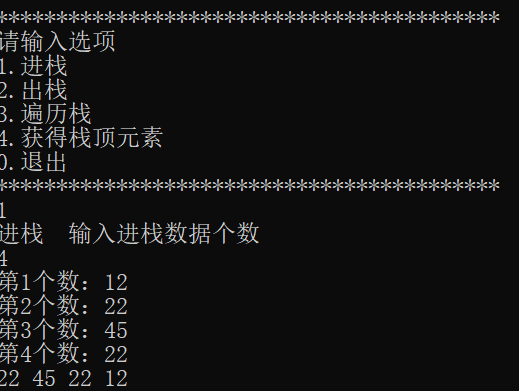
Stack s;   
 InitStack(&s);   
 int isquit = 0;   
 while(!isquit){   
   
 int a = 10;   
 int i =0;   
 int value = 0;   
 int num = 0;   
 printf("\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n请输入选项\n1.进栈\n2.出栈\n3.遍历栈\n4.获得栈顶元素\n0.退出\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");   
 scanf("%d",&a);   
 switch(a)   
 {   
 case 0: isquit = 1;   
 break;   
 case 1:printf("进栈 输入进栈数据个数\n");   
 scanf("%d",&num);   
 for (i = 0;i < num;i++)   
 {   
 printf("第%d个数：",i+1);   
 scanf("%d",&value);   
 Push(&s,value); // 调用Push函数   
 }   
 PrintStack(&s); // 调用遍历函数   
 break;   
 case 2:printf("出栈 输入出栈数据个数\n");   
 scanf("%d",&num);   
 for (i = 0;i < num;i++)   
 {   
 printf("第%d个数：%d\n",i+1,Pop(&s));   
 }   
 printf("现在栈元素为\n");   
 PrintStack(&s); // 调用遍历函数   
 break;   
 case 3:printf("遍历栈 现在栈中元素\n");   
 PrintStack(&s);   
 break;   
 case 4:printf("栈顶元素为%d\n",GetTop(&s));   
 break;   
 default:printf("重新输入");   
 }   
   
 }

1. **流程图**

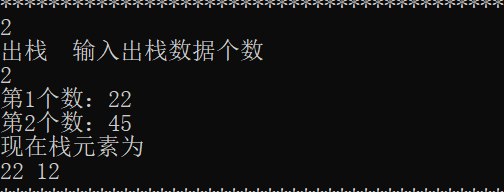


1. **测试与分析**

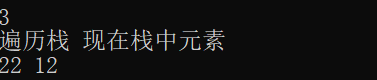
进栈



出栈



遍历栈



获得栈顶元素

退出

