**试题五（共15分）**

**阅读下列说明和C代码，将应填入 （n） 处的字句写在答题纸的对应栏内。**

**【说明】**

**栈（Stack）结构是计算机语言实现中的一种重要数据结构。对于任意栈，进行插入和删除操作的一端称为栈顶（Stack Top），而另一端称为栈底（Stack Bottom）。栈的基本操作包括：创建栈（NewStack）、 判断栈是否为空（IsEmpty）、判断栈是否已满（IsFull）、获取栈顶数据（Top）、压栈/入栈（Push）、弹栈/出栈（Pop）。**

**当设计栈的存储结构时，可以采取多种方式。其中，采用链式存储结构实现的栈中各数据项不必连续存储（如下图所示）。**

**以下C代码采用链式存储结构实现一个整数栈操作。**

**【C代码】**

**typedef struct List {  
int data; // 栈数据  
struct List\* next; // 上次入栈的数据地址**

**}List;**

**typedef struct Stack {  
List\* pTop; // 当前栈顶指针**

**}Stack;**

**Stack\* NewStack() { return (Stack\*)calloc(1,sizeof(Stack)); }**

**int IsEmpty(Stack\* S){ //判断栈S是否为空栈  
if(（1）) return 1;   
return 0;**

**}**

**int Top(Stack\* S){ //获取栈顶数据。若栈为空，则返回机器可表示的最小整数  
if( IsEmpty(S) ) return INT\_MIN;  
return （2） ;**

**}**

**void Push(Stack\* S, int theData) {//将数据theData压栈  
List\* newNode;  
newNode = (List\*)calloc(1, sizeof(List));  
newNode->data = theData;  
newNode->next = S->pTop;  
S->pTop = （3） ;**

**}**

**void Pop(Stack\* S) {//出栈  
List\* lastTop;  
if( IsEmpty(S) ) return;  
lastTop = S->pTop;  
S->pTop = （4） ;   
free(lastTop);**

**}**

**#define MD(a) a<<2**

**int main(){  
int i;  
Stack\* myStack;  
myStack = NewStack();  
Push(myStack, MD(1));  
Push(myStack, MD(2));  
Pop(myStack);  
Push(myStack, MD(3)+1);  
while( !IsEmpty(myStack) ){  
 printf("%d", Top(myStack));  
 Pop(myStack);**

**}  
return 0;**

**}**

**以上程序运行时的输出结果为： （5）**