**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）**

**试题五分析**

**本题考查基本程序设计能力。**

**堆栈是软件设计中常使用的一种经典数据结构，题目给出的操作都是任何堆栈都具有的基本操作。堆栈的存储结构通常采用数组或链表形式，但无论采用哪种存储结构，整体上呈现的是后进先出的特点，即后进入堆栈的元素先出栈。题目中给出的结构体Stack仅包含一个指向栈顶元素的指针（栈顶指针），当且仅当堆栈中没有元素时，该指针应为NULL。当向堆栈中增加元素时，首先需要动态创建该元素的存储区，并且栈顶指针指向该元素。当元素出栈时，栈顶指针则指向出栈元素的紧前一个元素。结构体List表示栈中元素，包含对应的数据和指向紧上次入栈的元素指针next，对于第1个入栈的元素，指针next为NULL,而其他元素中的指针next一定不为NULL。**

**C语言中，如果用一个整数型表达式表示条件判定语句的话，该表达式的值为0则表示假，非0表示真。从给定程序代码可以看出，对于函数IsEmpty，若其返回值为0则表示堆栈非空，否则表示堆栈为空。因此，对于空（1），必须填写可表示堆栈为空的判定语句：S==NULL||S->pTop==NULL，这2个条件中只要有1个条件满足，则表明堆栈S为空。对于空（2），此时需要返回栈顶元素中的数据，而栈顶元素为S->pTop，所以对应的数据应该为S->pTop->data。**

**对于压栈操作Push，在为新元素获取存储空间后，必须调整堆栈的栈顶指针S->pTop指向新元素的存储区，即S->pTop=newNode。对于弹栈操作Pop，弹出栈顶元素lastTop后，需要调整栈顶指针，使其指向被弹出元素的下一个元素，即S->pTop=S->pTop->next，或S->pTop=lastTop->next。**

**对于main函数中宏MD(x)，在程序预编译时会按字符替换为 "x<<2"。所以在main函数中，首先入栈的元素为"1<<2"，即整数4，第2个入栈的元素为"2<<2"，即整数8，其次将8弹出，然后再将"3<<2+1"入栈，C语言中 "+"优先级高于"<<"，所以此时入栈者为整数24，而此时堆栈中有2个元素，其中栈顶元素为24，下一元素为4。最后，若堆栈非空，则循环完成显示栈顶元素的值、弹出栈顶元素的操作，直至堆栈为空。所以程序执行时的输出内容为"24 4"。**

**参考答案**

**（1）S == NULL || S->pTop == NULL （2）S->pTop->data （3）newNode**

**（4）S->pTop->next，或 lastTop->next （5）24 4**

**[NextPage]**