1. typedef struct SNode \*PtrToSNode;
2. struct SNode {
3. ElementType Data;
4. PtrToSNode Next;
5. };
6. typedef PtrToSNode Stack;
8. Stack CreateStack( )
9. { /\* 构建一个堆栈的头结点，返回该结点指针 \*/
10. Stack S;
12. S = (Stack)malloc(sizeof(struct SNode));
13. S->Next = NULL;
14. return S;
15. }
17. bool IsEmpty ( Stack S )
18. { /\* 判断堆栈S是否为空，若是返回true；否则返回false \*/
19. return ( S->Next == NULL );
20. }
22. bool Push( Stack S, ElementType X )
23. { /\* 将元素X压入堆栈S \*/
24. PtrToSNode TmpCell;
26. TmpCell = (PtrToSNode)malloc(sizeof(struct SNode));
27. TmpCell->Data = X;
28. TmpCell->Next = S->Next;
29. S->Next = TmpCell;
30. return true;
31. }
33. ElementType Pop( Stack S )
34. { /\* 删除并返回堆栈S的栈顶元素 \*/
35. PtrToSNode FirstCell;
36. ElementType TopElem;
38. if( IsEmpty(S) ) {
39. printf("堆栈空");
40. return ERROR;
41. }
42. else {
43. FirstCell = S->Next;
44. TopElem = FirstCell->Data;
45. S->Next = FirstCell->Next;
46. free(FirstCell);
47. return TopElem;
48. }
49. }