```
typedef struct SNode *PtrToSNode;
struct SNode {
    ElementType Data;
    PtrToSNode Next;
};
typedef PtrToSNode Stack;

Stack CreateStack( )
{ /* 构建一个堆栈的头结点，返回该结点指针 */
    Stack S;

    S = (Stack)malloc(sizeof(struct SNode));
    S->Next = NULL;
    return S;
}

bool IsEmpty ( Stack S )
{ /* 判断堆栈S是否为空，若是返回true；否则返回false */
    return ( S->Next == NULL );
}

bool Push( Stack S, ElementType X )
{ /* 将元素X压入堆栈S */
    PtrToSNode TmpCell;

    TmpCell = (PtrToSNode)malloc(sizeof(struct SNode));
    TmpCell->Data = X;
    TmpCell->Next = S->Next;
    S->Next = TmpCell;
    return true;
}

ElementType Pop( Stack S )
{ /* 删除并返回堆栈S的栈顶元素 */
    PtrToSNode FirstCell;
    ElementType TopElem;

    if( IsEmpty(S) ) {
        printf("堆栈空");
        return ERROR;
    }
    else {
        FirstCell = S->Next;
        TopElem = FirstCell->Data;
        S->Next = FirstCell->Next;
        free(FirstCell);
        return TopElem;
    }
}
```