```
typedef int Position;
struct SNode {
    ElementType *Data; /* 存储元素的数组 */
    Position Top;      /* 栈顶指针 */
    int MaxSize;       /* 堆栈最大容量 */
};
typedef struct SNode *Stack;

Stack CreateStack( int MaxSize )
{
    Stack S = (Stack)malloc(sizeof(struct SNode));
    S->Data = (ElementType *)malloc(MaxSize * sizeof(ElementType));
    S->Top = -1;
    S->MaxSize = MaxSize;
    return S;
}

bool IsFull( Stack S )
{
    return (S->Top == S->MaxSize-1);
}

bool Push( Stack S, ElementType X )
{
    if ( IsFull(S) ) {
        printf("堆栈满");
        return false;
    }
    else {
        S->Data[++(S->Top)] = X;
        return true;
    }
}

bool IsEmpty( Stack S )
{
    return (S->Top == -1);
}

ElementType Pop( Stack S )
{
    if ( IsEmpty(S) ) {
        printf("堆栈空");
        return ERROR; /* ERROR是ElementType的特殊值，标志错误 */
    }
    else
        return ( S->Data[(S->Top)--] );
}
```