

## スタックの実現

```
typedef struct {
    int          top ;
    elementtype  elements[1..maxlength];
} STACK ;
```

この本では、配列 *elements* の添字 1 から用いることにしており、*top* は STACK に含まれる要素の個数と同じになっている。(そうではなく、*top* は次に要素を入れる配列 *elements* の添字を示すものとする 것도できる。その場合には、*top* は STACK に含まれる要素の個数より 1 だけ多くなる。)

```
void Makenull( S )
    入力 : S :    STACK(名前呼び)
    出力 : S:    STACK
{
    S.top := 0;
} /* Makenull */
```

なお、STACK *S* は構造体なので、C 言語ではポインタを渡さねばならない。従って、

```
void Makenull( STACK * ptr_S )
{
    (*ptr_S).top := 0;
} /* Makenull */
```

とし、これを呼び出す側では、

```
Makenull( &S );
```

とする。

```
elementtype TopElement( S )
    入力 : S :    STACK(値呼び)
    出力 : TopElement : elementtype
    /* ここでは、elementtype の要素を関数の戻し値とできるものとしている */
{
    if( 1 < S.top < maxlength ) return S.element[S.top];
    else エラーを出力する ;
} /* TopElement */
```

```

void Pop( S )
    入力 : S : STACK(名前呼び)
    出力 : S : STACK
{
    if( 1 < S.top < maxlength ) S.top := S.top - 1;
    else エラーを出力する ;
} /* Pop */

```

```

boolean Empty( S )
    入力 : S : STACK(値呼び)
    出力 : Empty : boolean
{
    if( 0 < S.top < maxlength ){
        if( S.top = 0 ) return true ;
        else return false ;
    }
    else エラーを出力する ;
} /* Empty */

```