

連結リストによるキューの実現

```
typedef struct {
    elementtype    element ;
    CELL↑         next ;      /* next は CELL へのポインタ */
} CELL ;
```

```
typedef struct {
    CELL↑         front ;
    CELL↑         rear ;     /* front, rear は CELL へのポインタ */
} QUEUE ;
```

これまでも書いてきたように、パラメタのキュー Q は、C 言語では Q へのポインタを渡すことになる。例えば、*Makenull* は C 言語では下のように書き、

```
position Makenull( QUEUE* ptr_Q ) { }
```

これを呼び出す側では、次のようにする。

```
Makenull( &Q );
```

```
void Makenull( Q )
    入力 : Q : QUEUE(名前呼び)
    出力 : Q : QUEUE
{
    /* Q はセルを持っていない状態であるとしている */
    Q.front := new( CELL );
    (↑Q.front).next := NULL ;
    Q.rear := Q.front ;
} /* Makenull */
```

```
elementtype FrntElmnt( Q )
    入力 : Q : QUEUE(値呼び)
    出力 : FrntElmnt : 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 ( (↑Q.front).next == NULL ) return (↑(↑Q.front).next).element ;
    else エラーを出力する ;
} /* FrntElmnt */
```

```

void Enqueue( x, Q )
    入力 : x : elementtype
           Q : Queue(名前呼び)
    出力 : Q : Queue
{
    ( $\uparrow$ Q.rear).next := new( CELL );
    Q.rear := ( $\uparrow$ Q.rear).next ;
    ( $\uparrow$ Q.rear).element := x ;
    ( $\uparrow$ Q.rear).next := NULL ;
} /* Enqueue */

```

```

boolean Empty( Q )
    入力 : Q : Queue(値呼び)
    出力 : Empty : boolean
{
    if ( Q.front == Q.rear ) return true ;
    else return false ;
} /* Empty */

```