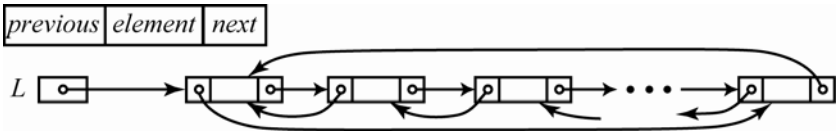
アルゴリズムとデータ構造1　　演習問題 (4)　　回答

整数の要素を *element* 欄に蓄えた巡回リスト（circular list）が次のように定義されている.

```
typedef  struct {
                int          element ;      /*  整数の要素を入れる欄  */
                DLCELL↑      previous ;     /*  1つ前のセルを指すポインタ  */
                DLCELL↑      next ;         /*  次のセルを指すポインタ  */
        } DLCELL ;
typedef         DLCELL↑      DCposition ;   /*  DCposition は DLCELL 型データへのポインタ型  */

DCposition      CL1 ;                       /*  巡回リスト CL1 は DLCELL 型データを指すポインタ  */
DCposition      CL2 ;                       /*  巡回リスト CL2 も DLCELL 型データを指すポインタ  */

/*  巡回リスト CL1 および CL2 は, element 欄に意味の無いデータが入っているヘッダセルを指す  */
/*  ヘッダセルの previous は最後のセルを指し, 最後のセルの next はヘッダセルを指す  */
/*  巡回リストが要素を持たないとき, ヘッダセルの previous および next は共にヘッダセルを指す  */
```

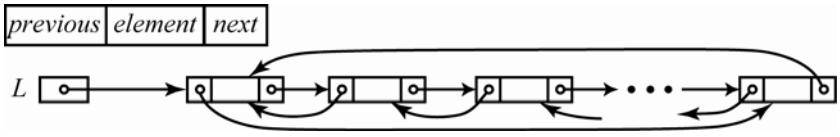



```
int   CheckCList(  int   x,   DCposition   L   )
    /*   巡回リスト L の中に整数 x が含まれていたら 1（true）を，さもなくば 0（false）を返す関数   */
    /*   x は整数で値呼び，L は巡回リストで値呼び   */
{
      DCposition   p ;                          /*  p は DLCELL へのポインタで局所変数  */

      p  :=  (↑L).next ;                        /*  p は先頭の要素が入った DLCELL 型のセルを指す  */
      while (   p ≠ L    ) {                   /*  未だ調べていない L の要素がある間繰り返す  */
            if (   (↑p).element = x    )
                  return   1 ;                  /*  x が見出されたので, 1 を返してこの関数から抜ける  */
            else      p  :=  (↑p).next ;
      }
      return   0 ;                              /*  x は見つからなかったので, 0 を返してこの関数から抜ける  */
} /*   CheckCList  */
```

```
void DeleteLastDLCell( DCposition L )
    /* 巡回リスト L の最後の要素を取り除く */
{
    DCposition    q ;                          /* DLCELL 型データへのポインタで局所変数 */

    q := (↑L).previous ;                       /* q は最後の要素が入った DLCELL 型のセルを指す */
    if ( q = L ) {
        "連結リストは空であると出力する" ;
    }
    else {
        (↑L).previous := (↑q).previous ;       /* ヘッダセルの previous が除去後最後になるセルを指す */
        (↑(↑q).previous).next := L ;           /* 除去後最後になるセルの next がヘッダセルを指す */
        free( q ) ;                            /* q が指す最後のセル free する */
    }
} /* DeleteFirstDLCell */
```




```
void DeleteFirstDLCell( DCposition L )
    /* 巡回リスト L の先頭の要素を取り除く */
{
    DCposition    q ;                          /* DLCELL 型データへのポインタで局所変数 */

    q := (↑L).next ;                           /* q は先頭の要素が入った DLCELL 型のセルを指す */
    if ( q = L ) {
        "連結リストは空であると出力する" ;
    }
    else {
        (↑q).previous := (↑L).previous ;       /* q が指すセルの previous が最後のセルを指す */
        (↑(↑L).previous).next := q ;           /* q が指すセルを最後のセルの next が指す */
        free( L ) ;                            /* これまでのヘッダセルを free する */
        L := q ;                               /* q が指すセルをヘッダセルにする */
    }
} /* DeleteFirstDLCell */
```