

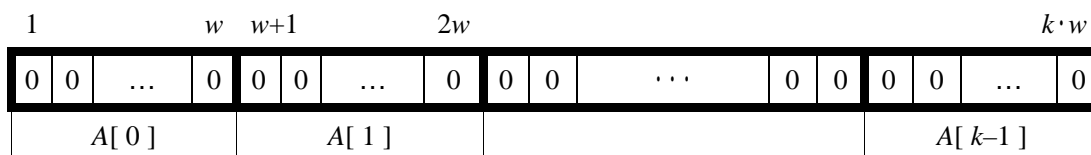
ビットベクトルによる集合の実現(C 言語プログラム)

```
long int  A[ k-1 ], B[ k-1 ], C[ k-1 ] ;          /* k は別のところで定義された定数である */
          /* 配列 A[ k-1 ], B[ k-1 ] および C[ k-1 ] はそれぞれ集合 A, B および C を表す */
```

```
void Intersection( long int  A[ ], B[ ], C[ ] )
/* 長さが k の long int 型の配列を用いて表された集合 A[ ] および B[ ] の */
/* 共通集合 C[ ] を同じデータ構造で作る */
{
    for ( i=0; i<k; i++ )          C[i] = A[i]&B[i] ;
        /* 0 < i < k なる各 i に対して, ビット毎の論理積をとり, C[i] とする */
} /* Intersection */
```

```
void Makenull( long int  A[ ] )
/* 長さが k の long int 型の配列を用いて表された集合 A[ ] を空にする */
{
    for ( i=0; i<k; i++ )          C[i] = C[i]^C[i] ;
        /* 0 < i < k なる各 i に対して, ビット毎の排他的論理和をとり, C[i] とする */
        /* 排他的論理和(exclusive or)は, 0^1 と 1^0 のときのみ 1 となる演算 */
} /* Makenull */
```

以下では, long int 型の整数が w ビットからなるものとし, $A[0]$ の最上位ビット(左端)が要素番号 1 に対応したビット, $A[0]$ の最下位ビット(右端)が要素番号 w に対応したビット, $A[1]$ の最上位ビットが要素番号 $w+1$ に対応したビット, $A[k-1]$ の最下位ビットが要素番号 $k \cdot w$ に対応したビットとする。(下図参照)



また,

```
#define false 0 ;
```

```
#define true 1 ;
```

と定義してあるものとする.

```
boolean Member( int x, long int A[ ] )
/* 要素番号 x の要素が集合 A に含まれるか否かを判定する関数 */
/* x ∈ A であれば true, さもなくば false を返す */
/* 定数 w は long int 型のビット数, maxlength は普遍集合 U の要素の個数 */
/* ErrorFunction は要素番号が異常であることを表示する関数 */
{
    int i, d ;
    long int mask ;
    if ( 0 < x <= maxlength ) {          /* 0 < x <= maxlength であるならば */
        i = x/w ;                        /* i は x を w で割った商(整数) */
        d = x - i*w ;                    /* d は x を w で割った余り(整数) */
        if ( d = 0 ) {
            i = i - 1 ;                  /* x は A[i - 1] に含まれるビットに対応 */
            mask = 1 ;                  /* mask は, 右端のビットが 1 で */
                                        /* 他のビットは 0 の long int 型変数 */
        }
        else    mask = 1 << w - d ;     /* mask は, 左端から d ビット目が 1 で */
                                        /* 他のビットは 0 の long int 型変数 */
        if ( mask & A[i] = 0 )          /* A[i] の左端から d ビット目が 1 でない */
            return false ;
        else    return true ;
    }
    else    ErrorFunction ;
} /* Member */
```