

他と異なる数を偽物 (*Fake*) と呼ぼう.

```
int FindFake3 ( x, y, z, A, flag )
    入力:   x, y, z :   正整数 (値呼び)
           A   :   実数値配列
           flag :   文字 (値呼び)   /* flag = "small" : 偽物は小さい */
                                           /* flag = "large" : 偽物は大きい */

    出力:   FindFake3 :   正整数 /* x, y, z の中の偽物 */
/* A[x], A[y], A[z] の中から 1 回の比較で偽物を見出す */
/* ただし, 偽物が本物より大きいか小さいか分かっている */
{
    if ( A[x] < A[y] ) {
        if ( flag = "small" )      return x;      /* A[x] が偽物 */
        else                       return y;      /* A[y] が偽物 */
    }
    else if ( A[x] > A[y] ) {
        if ( flag = "small" )      return y;      /* A[y] が偽物 */
        else                       return x;      /* A[x] が偽物 */
    }
    else /* A[x] = A[y] */         return z;      /* A[z] が偽物 */
} /* FindFake3 */
```

```
int FindFake4 ( x, y, z, w, A )
    入力:   x, y, z, w :   正整数 (値呼び)
           A   :   実数値配列

    出力:   FindFake4 :   正整数 /* x, y, z, w の中の偽物 */
/* A[x], A[y], A[z], A[w] の中から 2 回の比較で偽物を見出す */
{
    if ( A[x] < A[y] ) { /* A[x], A[y] のどちらかが偽物 */
        if ( A[y] > A[z] ) /* A[y] が大きい偽物 */
            return y;
        else /* A[y] = A[z] である . A[x] < A[y] < A[z] は起こらない */
            return x; /* A[x] が小さい偽物 */
    }
    else if ( A[x] > A[y] ) { /* A[x], A[y] のどちらかが偽物 */
        if ( A[y] < A[z] ) /* A[y] が小さい偽物 */
            return y;
        else /* A[y] = A[z] である . A[z] < A[y] < A[x] は起こらない */
            return x; /* A[x] が大きい偽物 */
    }
    else { /* A[x] = A[y] である . A[z], A[w] のどちらかが偽物 */
        if ( A[y] < A[z] ) /* A[z] が大きい偽物 */
            return z;
        else if ( A[y] > A[z] ) /* A[z] が小さい偽物 */
            return z;
        else /* A[y] = A[z] である . A[w] が偽物 . その大小は不明 */
            return w;
    }
} /* FindFake4 */
```

```

int FindFake12 ( A )
    入力:  A[1..12] :      実数値配列
    出力:  FindFake12 :   1 ~ 12 の正整数
/* A[1] ~ A[12] の中から 3 回の比較で偽物を見出す */
{
    A1_4 := A[1]+A[2]+A[3]+A[4];
    A5_8 := A[5]+A[6]+A[7]+A[8];
    if ( A1_4 > A5_8 ) {
        /* A[1] ~ A[8] の中に偽物 */
        B := A[1]+A[2]+A[3]+A[5]+A[6];
        C := A[9]+A[10]+A[11]+A[12]+A[4];
        if ( B > C )
            /* A[1] ~ A[3] の中に大きい偽物 */
            return FindFake3 ( 1, 2, 3, A, "large" );
        else if ( B < C ) {
            /* A[4] が大きい偽者か, あるいは */
            /* A[5], A[6] のどちらかが小さい偽物 */
            if ( A[5] < A[6] )
                /* A[5] が小さい偽物 */
                return 5;
            else if ( A[5] > A[6] )
                /* A[6] が小さい偽物 */
                return 6;
            else
                /* A[5] = A[6] である . A[4] が大きい偽者 */
                return 4;
        }
        else
            /* B = C である . A[7], A[8] のどちらかが小さい偽物 */
            return FindFake3 ( 6, 7, 8, A, "small" );
    }
    else if ( A1_4 < A5_8 ) {
        /* A[1] ~ A[8] の中に偽物 */
        B := A[1]+A[2]+A[3]+A[5]+A[6];
        C := A[9]+A[10]+A[11]+A[12]+A[4];
        if ( B < C )
            /* A[1] ~ A[3] の中に小さい偽物 */
            return FindFake3 ( 1, 2, 3, A, "small" );
        else if ( B > C ) {
            /* A[4] が小さい偽者か, あるいは */
            /* A[5], A[6] のどちらかが大きい偽物 */
            if ( A[5] > A[6] )
                /* A[5] が大きい偽物 */
                return 5;
            else if ( A[5] < A[6] )
                /* A[6] が大きい偽物 */
                return 6;
            else
                /* A[5] = A[6] である . A[4] が小さい偽者 */
                return 4;
        }
        else
            /* B = C である . A[7], A[8] のどちらかが小さい偽物 */
            return FindFake3 ( 6, 7, 8, A, "small" );
    }
    else
        /* A1_4 < A5_8 である . A[9] ~ A[12] の中に偽物 */
        return FindFake4 ( 9, 10, 11, 12, A );
}
/* FindFake12 */

```