

**Standard C Function: Mod-2 Polynomial Modular Checksum**

```

int mod2polychecksum( int msgbit[], int numbits,
                      int mod2poly[], int deg ) {
    int checksum, d, n;
    for(n=0; n<numbits-deg; n++ )
        if( msgbit[n]==1 )
            for(d=0; d<=deg; d++)
                msgbit[n+d] ^= mod2poly[deg-d];
    for(checksum=0; n<numbits; n++)
        checksum = 2*checksum + msgbit[n];
    return checksum;
}

```

We will use `mod2poly[]={1,1,0,1}`, representing  $1+t+t^3$  with the coefficient of  $t^3$  at index 3. Applying this with `msgbit[]` containing the encoded bits from the message “Elephants are approaching from the South!” as computed in Solution 20, and putting the left-most bit of “E” at index 0, yields a checksum of 0.

Calling `mod2polychecksum()` with the “...North!” base-2 bits in the array `msgbit[]` and the same mod-2 modulus polynomial yields a checksum of 1, distinguishing the strings. However, “Elephants are approaching from the NORTH!” yields the following bitstring:

```

11000101 01101100 01100101 11110000 11101000 11100001 11101110 01110100
11110011 10100000 11100001 01110010 01100101 10100000 11100001 11110000
11110000 01110010 01101111 11100001 01100011 11101000 01101001 11101110
11100111 10100000 01100110 01110010 01101111 11101101 10100000 01110100
11101000 01100101 10100000 01001110 01101111 01110010 01110100 11101000
00100001 (base 2)

```

Then `mod2polychecksum()` returns its checksum as 0. Thus, this message is not distinguished from the first by its 3-bit checksum.  $\square$