Problem 4—The Hamming Distance Problem

The Hamming distance between two strings of bits (binary integers) is the number of corresponding bit positions that differ. This can be found by using XOR on corresponding bits or equivalently, by adding corresponding bits (base 2) without a carry. For example, in the two bit strings that follow:

A	0 1 0 0 1 0 1 0 0 0
B	1 1 0 1 0 1 0 1 0 0
A XOR B =	1 0 0 1 1 1 1 1 0 0

The Hamming distance (*H*) between these 10-bit strings is 6, the number of 1's in the XOR string. Given a length *N* and a Hamming distance *H*, you are to print A list of all possible bit strings of length *N* that are Hamming distance *H* from the bit string containing all 0's (origin). That is, you are to print all bit strings of length *N* with exactly *H* 1's.

INPUT SPECIFICATION.

Each input case will be in the form "*N* <exactly one space> *H* **<EOLN>**" The last input case will be followed by "0 0**<EOLN>**". The 0 0 case is not to processed; it merely specifies the end of input. For all input cases, $1 \le H \le N \le 16$.

OUTPUT SPECIFICATION.

The output cases should appear in the same order as their respective input cases. Each output case should begin with "Case *C*" (where *C* is the case number) and two **<EOLN>**'s. Then each valid bit string should be printed *in sorted numerical order*, each one followed by **<EOLN>**. An extra **<EOLN>** should be printed after each data case.

SAMPLE INPUT.

```
4·2<EOLN>
5·1<EOLN>
0·0<EOLN>
<EOF>
```

SAMPLE OUTPUT.

```
Case·1<EOLN>
<EOLN>
0011<EOLN>
0101<EOLN>
0110<EOLN>
1001<EOLN>
1010<EOLN>
1100<EOLN>
<EOLN>
Case·2<EOLN>
<EOLN>
00001<EOLN>
00010<EOLN>
00100<EOLN>
01000<EOLN>
10000<EOLN>
<EOLN>
<EOF>
```