

# Problem A - QR Code decoder!

---

## Description

QR (Quick Response) codes are two-dimensional barcodes that are utilized for storing and easily accessing data. The codes are made up mostly of black and white squares, and can be scanned using a smartphone and a QR code scanning app. These codes are often used in various industries, such as advertising, events, and product labeling, due to their fast scanning ability and significant storage capacity. In fact, they provide a quick and practical way of transferring information.



Figure 1 - General QR Code.

Each QR code can be seen as a matrix of binary cells, meaning that each cell can be represented as a black or white pixel/square.

In order to send and store QR codes, compressing QR codes is essential to save network traffic and storage space. So, a tech start up - QR Shorter Inc - is developing a method to compress the QR codes. The method encodes the QR code as summary of its characteristics.

The encoding algorithm goes as follows:

- save the information about the size of the QR Code, meaning the number of cells per line/column ( $N$ ). The QR Shorter Inc. works only with squared QR Codes.
- save the amount of black cells per line ( $lb$ , line blacks, array of size  $N$ )
- save the amount of black cells per column ( $cb$ , column blacks, array of size  $N$ )
- save the amount of color transitions per line ( $lt$ , line color transitions, array of size  $N$ )
- save the amount of color transitions per column ( $ct$ , column color transitions, array of size  $N$ )
- save the amount of black cells per quadrant ( $qb$ , quadrant blacks, array of size 4)
- save the amount of black cells per diagonal - main and antidiagonal - ( $db$ , diagonal blacks, array of size 2)

### Encoding Notes:

Consider the QR Code as a squared matrix of side  $N$ . For a matrix cell  $c$  positioned in the line  $l \leq l \leq N$  and a column  $l \leq c \leq N$ ,  $c$  belongs to:

- quadrant 1 if  $l \leq \text{floor}(N/2)$  and  $c > \text{floor}(N/2)$ ,
- quadrant 2 if  $l \leq \text{floor}(N/2)$  and  $c \leq \text{floor}(N/2)$ ,
- quadrant 3 if  $l > \text{floor}(N/2)$  and  $c \leq \text{floor}(N/2)$ ,
- quadrant 4 if  $l > \text{floor}(N/2)$  and  $c > \text{floor}(N/2)$ ,
- diagonal 1 (main diagonal) if  $l = c$
- diagonal 2 (antidiagonal) if  $L-l+l = c$

Figure 2 shows two examples on how the encoding is performed.

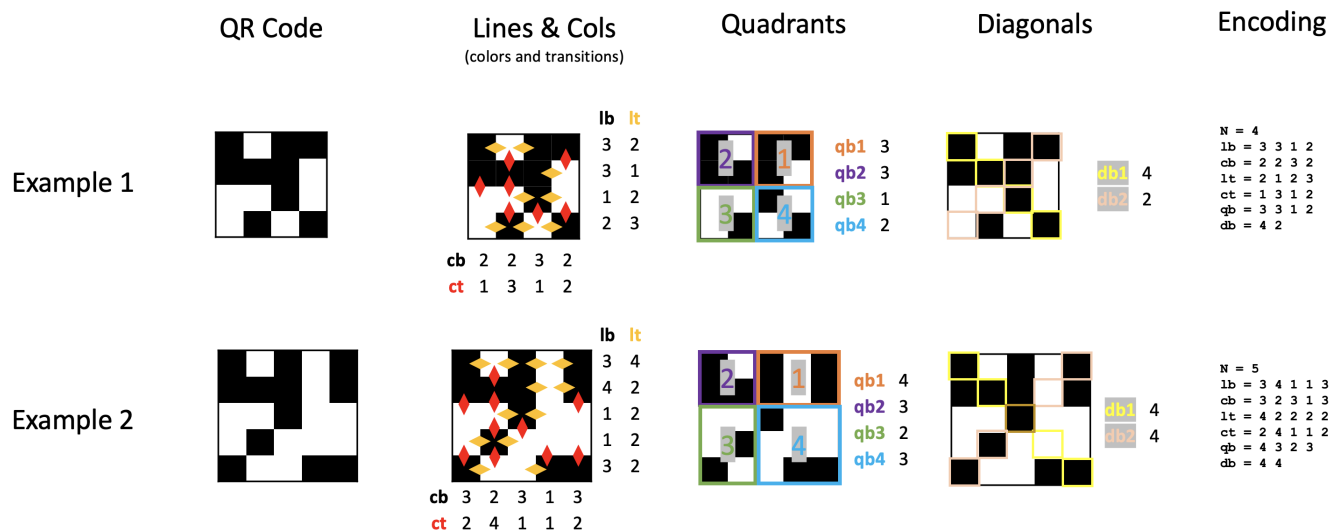


Figure 2 - Encoding process of two example QR Codes.

From that information, it should be possible to reconstruct (decode) the QR code and save some space. The storage needed for that info is  $1+4*N+4+2$ , which is smaller than  $N*N$ , for every  $N > 5$ .

### Example:

From the encoding

```

N = 4
lb = 1 1 1 1
cb = 1 1 1 1
lt = 1 2 2 1
ct = 1 2 2 1
qb = 0 2 0 2
db = 4 0

```

the following QR code can be decoded:

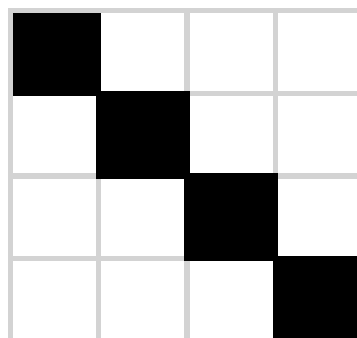


Figure 3 - QR Code decoded from the encoded information.

### Oh no, a problem!

The company found out that, in some cases, two (or more) different QR codes generate the same encoding. See the example in Figure 4.

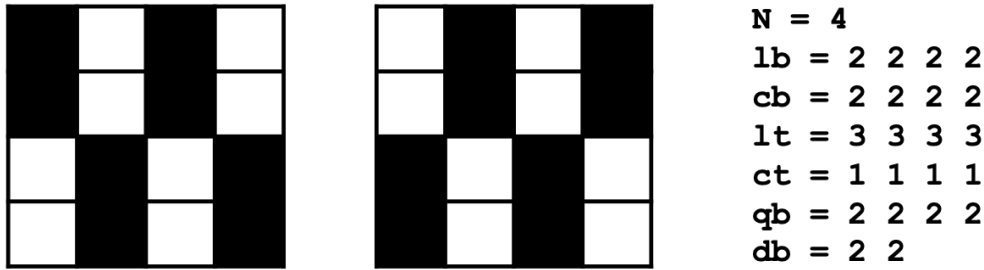


Figure 4 - Two QR codes which generate the same encoding.

This is a big step back in their business model. They decided that, instead of withdrawing the idea, they would implement a module that evaluates each encoding and says if it decodes into a single QR code or to multiple QR codes. If more than one QR code can be decoded from that encoding, it will be considered an invalid encoding.

Your job is to create this module. Given an encoding, find out how many QR codes can be generated from it. If only one, print the decoded QR code, if more than one, state the encoding as INVALID and tell how many QR codes it can generate. Additional, there might be some defects in the encodings, so if you find one of those cases that cannot generate any QR coded state as DEFECT.

## Input

The input starts with one line containing the number of encodings to be assessed.

Then, for each encoding, the first line contains the size of the QR code in lines/columns:  $2 \leq N \leq 30$

After that:

- One line containing  $N$  numbers (each between 1 and  $N$ ), corresponding to the number of black cells in each line  $l$  to  $N$  ( $lb$ ).
- One line containing  $N$  numbers (each between 1 and  $N$ ), corresponding to the number of black cells in each column  $l$  to  $N$  ( $cb$ ).
- One line containing  $N$  numbers (each between 1 and  $N-1$ ), corresponding to the number of color transitions in each line  $l$  to  $N$  ( $lt$ ).
- One line containing  $N$  numbers (each between 1 and  $N-1$ ), corresponding to the number of color transitions in each column  $l$  to  $N$  ( $ct$ ).
- One line containing 4 numbers (each between 1 and  $\text{floor}(N/2+1)^2$ ), corresponding to the number of black cells in each quadrant  $l$  to 4 ( $qb$ ).
- One line containing 2 numbers (each between 1 and  $N$ ), corresponding to the number of black cells in each diagonal  $l$  to 2 ( $db$ ).

## Output

For each test case, you should print one of the following possible outputs:

### 1. Valid case: 1 QR code decoded

If only one solution can be generated from the encoding, you should print the following phrase:

VALID: 1 QR Code generated.

After that, print the QR Code generated. Print every white cell as a space character ' ' and every black cell as a cardinal '#' character. Print a frame for the QR code using the character '+' for the corners, '-' for the top and bottom borders and '|' for the left and right borders.

### 2. Invalid case: more than one QR code decoded

If more than one QR Code is decoded, you should print the phrase (being  $k$  the number of QR Codes generated):

INVALID:  $K$  QR Codes generated.

3. **Defect case:** 0 QR codes decoded

If no QR Code can be decoded from the encoding, than print the phrase:

DEFECT: No QR Code generated.

---

**Example**

**Example input:**

```
3
3
1 0 1
1 0 1
1 0 1
1 0 1
0 1 0 1
2 0
4
2 2 2 2
2 2 2 2
3 3 3 3
1 1 1 1
2 2 2 2
2 2
4
2 3 4 3
2 3 4 3
1 1 0 1
1 1 0 1
4 3 4 1
2 4
```

---

**Example output:**

```
VALID: 1 QR Code generated!
+---+
|#  |
|  |
|  #|
+---+
INVALID: 2 QR Codes generated!
DEFECT: No QR Code generated!
```

---