A biologist is experimenting with DNA modification of bacterial colonies being grown in a linear array of culture dishes. By changing the DNA, he is able "program" the bacteria to respond to the population density of the neighboring dishes. Population is measured on a four point scale (from 0 to 3). The DNA information is represented as an array *DNA*, indexed from 0 to 9, of population density values and is interpreted as follows:

- In any given culture dish, let K be the sum of that culture dish's density and the densities of the dish immediately to the left and the dish immediately to the right. Then, by the next day, that dish will have a population density of *DNA[K]*.
- The dish at the far left of the line is considered to have a left neighbor with population density 0.
- The dish at the far right of the line is considered to have a right neighbor with population density 0.

Now, clearly, some DNA programs cause all the bacteria to die off (e.g., [0,0,0,0,0,0,0,0,0,0]). Others result in immediate population explosions (e.g., [3,3,3,3,3,3,3,3,3,3]). The biologist is interested in how some of the less obvious intermediate DNA programs might behave.

Write a program to simulate the culture growth in a line of 40 dishes, assuming that dish 20 starts with a population density of 1 and all other dishes start with a population density of 0.

Input

The input begins with a single positive integer on a line by itself indicating the number of the cases following, each of them as described below. This line is followed by a blank line, and there is also a blank line between two consecutive inputs.

For each input set your program will read in the DNA program (10 integer values) on one line.

Output

For each test case, the output must follow the description below. The outputs of two consecutive cases will be separated by a blank line.

For each input set it should print the densities of the 40 dishes for each of the next 50 days. Each day's printout should occupy one line of 40 characters. Each dish is represented by a single character on that line. Zero population densities are to be printed as the character ' '. Population density 1 will be printed as the character '`.`'. Population density 2 will be printed as the character '`x`'. Population density 3 will be printed as the character '`W`'.

Note: Whe show only the first ten lines of output (the total number of lines must be 50) and the spaces have been replaced with the character '`b`' for ease of reading. The actual output file will use the ASCII-space character, not '`b`'.

Sample Input

```
1

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

Sample Output

```
bbbbbbbbbbbbbbbbbbb.bbbbbbbbbbbbbbbbbbbb
bbbbbbbbbbbbbbbbbb...bbbbbbbbbbbbbbbbbbb
bbbbbbbbbbbbbbbbb.xbx.bbbbbbbbbbbbbbbbbb
bbbbbbbbbbbbbbbb.bb.bb.bbbbbbbbbbbbbbbbb
bbbbbbbbbbbbbbb.........bbbbbbbbbbbbbbbb
bbbbbbbbbbbbbb.xbbbbbbbx.bbbbbbbbbbbbbbb
bbbbbbbbbbbbb.bbxbbbbbxbb.bbbbbbbbbbbbbb
bbbbbbbbbbbb...xxxbbbxxx...bbbbbbbbbbbbb
bbbbbbbbbbb.xb.WW.xbx.WW.bx.bbbbbbbbbbbb
bbbbbbbbbb.bbb.xxWb.bWxx.bbb.bbbbbbbbbbb
```