Problem A. Red-blue table

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	256 megabytes

Aidos and Tima are going to play an interesting game on a table of sizes $N \times M$. Also, they have an unlimited number of stones of two colors: red and blue. They want to fill the entire table in such a way that each cell of the table contains exactly one stone.

Aidos likes rows of the table with the number of red stones strictly greater than the number of blue stones. Let's denote the number of these rows as A.

Tima likes columns of the table with the number of blue stones strictly greater than the number of red stones. Let's denote the number of these columns as B.

As they are given only one table, they decided not to annoy each other and fill the table such that the total number of rows that Aidos likes and columns that Tima likes would be as many as possible.

Formally, they will try to maximize the value of the expression $A + B$.

Help the guys to fill the table.

Input

The first line contains a single integer T ($1 \le T \le 1000$) — the number of tests.

Next T lines contain two integers numbers N, M ($1 \le N, M \le 1000$). It is guaranteed that the sum of $N \cdot M$ over all the tests will not exceed 10^6.

Output

The answer for each test consists of $N + 1$ lines. Output the maximum value of $A + B$ in one line. In each of the next N lines output M symbols ('+' — for a red stone, '-' — for a blue stone). If there are several solutions you can output any of them.

Scoring

This task contains six substasks:

1. $1 \le T \le 16, 1 \le N, M \le 4$. Scored 17 points.
2. $1 \le T \le 1000, 1 \le N, M \le 50, min(N, M) \le 3$. Scored 10 points.
3. $1 \le T \le 1000, 1 \le N, M \le 50, min(N, M) \le 5$. Scored 16 points.
4. $1 \le T \le 1000, 1 \le N, M \le 1000$. N and M — odd numbers. Scored 11 points.
5. $1 \le T \le 1000, 1 \le N, M \le 1000, N = M$. Scored 15 points.
6. $1 \le T \le 1000, 1 \le N, M \le 1000$. Scored 31 points.

Example

standard input	standard output
2 1 3 3 3	3 --- 4 +-+ +-+ +++