C. Help Caretaker

Autumn came late to the kingdom of Far Far Away. The harvest was exuberant and it is now time to get ready for the winter. As most people celebrate the Harvest festival, Simon the Caretaker tries to solve a very non-trivial task of how to find place for the agricultural equipment in the warehouse.

He's got problems with some particularly large piece of equipment, which is, of course, turboplows. The problem is that when a turboplow is stored, it takes up not some simply rectangular space. It takes up a T-shaped space like on one of the four pictures below (here character "#" stands for the space occupied by the turboplow and character "." stands for the free space):

```
###        ..#        .#.        #..
.#.        ###        .#.        ###
.#.        ..#        ###        #..
```

Simon faced a quite natural challenge: placing in the given $n \times m$ cells warehouse the maximum number of turboplows. As one stores the turboplows, he can rotate them in any manner (so that they take up the space like on one of the four pictures above). However, two turboplows cannot "overlap", that is, they cannot share the same cell in the warehouse.

Simon feels that he alone cannot find the optimal way of positioning the plugs in the warehouse that would maximize their quantity. Can you help him?

Input

The only line contains two space-separated integers n and m — the sizes of the warehouse ($1 \le n, m \le 9$).

Output

In the first line print the maximum number of turboplows that can be positioned in the warehouse. In each of the next n lines print m characters. Use "." (dot) to mark empty space and use successive capital Latin letters ("A" for the first turboplow, "B" for the second one and so on until you reach the number of turboplows in your scheme) to mark place for the corresponding turboplows considering that they are positioned in the optimal manner in the warehouse. The order in which you number places for the turboplows does not matter. If there are several optimal solutions for a warehouse of the given size, print any of them.

Examples

input
3 3

output
1 AAA .A. .A.

input
5 6

output
4 A..C.. AAAC.. ABCCCD

```
.B.DDD
BBB..D
```

```
.B.DDD
BBB..D
```

input

2 2

output

0
..
..