

The parks in the City of Logic are always a grid of $N \times N$ squares ($2 \leq N \leq 100$), where each square has one of the first 10 ASCII letters, `abcdefghijklmnopqrstuvwxyzABCDEFGHIJ`, in either lowercase or uppercase format. People from the City of Logic proudly follow only consistent paths when crossing parks. For example, if they step over a lowercase `c`, they will not allow themselves stepping over an uppercase `C` afterwards. To define this more precisely, a consistent path is a sequence of squares satisfying: consecutive squares are orthogonally adjacent; no letter occurs in both lowercase and uppercase format. That is to say, either a letter is not in the sequence at all, or it occurs only in lowercase, or only in uppercase format.

DdaAaA	D.....
CBAcca	C.....
eEaeeE	e.....
bBbabB	b.bab.
DbDdDc	DbD.D.
fFaAaCaC

You have to write a program to help the people from the City of Logic to find the length a shortest consistent path between the square with coordinates $(1, 1)$, in the upper left corner, and the square with coordinates (N, N) , in the lower right corner. For the example park above, the shortest consistent path has length 13.

Input

The input contains several test cases. The first line of a test case has a integer N ($2 \leq N \leq 100$), the size of the park. The next N lines contain, each one, a sequence of N letters, defining the park.

Output

For each test case in the input your program must output one line containing one integer, the length of a shortest consistent path. If there is no consistent path, output `-1`.

Sample Input

```
6
DdaAaA
CBAcca
eEaeeE
bBbabB
DbDdDc
fFaAaC
7
aAaaaaa
aAaaaAa
aAaaaAA
aaAaAaa
AaAaaAa
aaAAaAa
aaaaaAa
```

Sample Output

```
13
-1
```