

(‘0’s and ‘1’s in this table except those in the first row represent tasks 0 and 1, respectively, and ‘C’ means the conflict.)

Your job is to write a program that reports the minimum number of clock cycles in which the given pipeline can process 10 tasks.

Input

The input consists of multiple data sets, each representing the reservation table of a pipeline. A data set is given in the following format.

```
n
x0,0 x0,1 ... x0,n-1
x1,0 x1,1 ... x1,n-1
x2,0 x2,1 ... x2,n-1
x3,0 x3,1 ... x3,n-1
x4,0 x4,1 ... x4,n-1
```

The integer $n (< 20)$ in the first line is the width of the reservation table, or the number of clock cycles that is necessary to perform a single task. The second line represents the usage of unit0, the third line unit1, and so on. $x_{i,j}$ is either ‘X’ or ‘.’. The former means *reserved* and the latter *free*. There are no spaces in any input line. For simplicity, we only consider those pipelines that consist of 5 function units. The end of the input is indicated by a data set with 0 as the value of n .

Output

For each data set, your program should output a line containing an integer number that is the minimum number of clock cycles in which the given pipeline can process 10 tasks.

Note: In this sample case, it takes 41 clock cycles to process 10 tasks if each task is started as early as possible under the condition that it never conflicts with any previous tasks being processed.

```
      | 00000000001111111111222222222233333333334
clock | 01234567890123456789012345678901234567890
-----
unit0 | 0.1.00112.3.22334.5.44556.7.66778.9.8899.
unit1 | .0.1.....2.3.....4.5.....6.7.....8.9.....
unit2 | ..0.1.....2.3.....4.5.....6.7.....8.9....
unit3 | ...0.1.....2.3.....4.5.....6.7.....8.9...
unit4 | .....0.1.....2.3.....4.5.....6.7.....8.9
```

(The digits in the table except those in the clock row represent the task number.)

However, it takes only 34 clock cycles if each task is started at every third clock.

```
      | 0000000000111111111122222222223333
clock | 0123456789012345678901234567890123
-----
unit0 | 0..100211322433544655766877988.99.
unit1 | .0..1..2..3..4..5..6..7..8..9.....
unit2 | ..0..1..2..3..4..5..6..7..8..9....
unit3 | ...0..1..2..3..4..5..6..7..8..9...
unit4 | .....0..1..2..3..4..5..6..7..8..9
```

(The digits in the table except those in the clock row represent the task number.)

This is the best possible schedule and therefore your program should report 34 in this case.

Sample Input

```
7
X...XX.
.X.....
..X.....
...X...
.....X
0
```

Sample Output

```
34
```