Fall 2021
ECE30017-01 Problem Solving through Computational Thinking

Week 7

- C5. Polyomino Puzzle
  Deadline: 11:59 PM, 15 October (Fri)
- P5. Electricity Poles
  Deadline: 11:59 PM, 19 October (Tue)

C5. Polyomino Puzzle

A polyomino is a rectilinear shape obtained by combining one or more 1x1 square blocks. For instance, Figure 1 shows four polyominos with different combinations of such blocks. Suppose that we are playing with a puzzle to form a square with given polyominos. In this puzzle, each polyomino is not allowed to rotate, and all polyominos are required to be used to form a square. Figure 2 shows how we can achieve a square by arranging the four polyominos in Figure 1.

Write a program that find a solution of this puzzle for given polyominos.

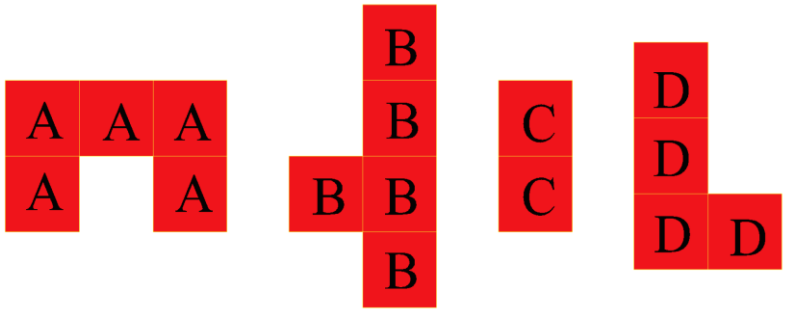


Figure 1. Four polyominos

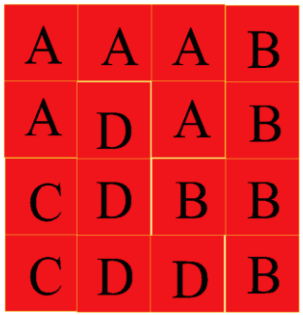


Figure 2. A square composed of the four polyominos in Figure 1.

(Continued)

Requirements

Input Data

- The first line from the standard input has an integer n, the number of polyominos in this puzzle for $1 \leq n \leq 5$. The following lines define the shapes of n polyominos. These polyominos in sequence have IDs from 1 to n, respectively.
- For each polyomino n, $1 \leq n \leq 5$, the first line gives two numbers h and w which represent the height and the width of a polyominos respectively, for $1 \leq h \leq 4$ and $1 \leq w \leq 4$. Then, h lines follow, each of which contains a number consisting of w binary digits, which defines the shape of a polyomino. A binary digit is 1 if and only if the polyomino has a square block at the corresponding position.

Output Data

- Print out an array that represents the arrangement of the given polyominos to the standard output. The array should be a square, that is, the number of rows should be the same as that of the columns. A square block in the array should be specified by the ID of the polyomino at the corresponding position. If it is impossible to form a square array, then print "No solution possible" as the output.
- Your program must return the result within 3.0 seconds

Test Case Example

Input data

```
4
2 3
1 1 1
1 0 1
4 2
0 1
0 1
1 1
0 1
2 1
1
1
3 2
1 0
1 0
1 1
```

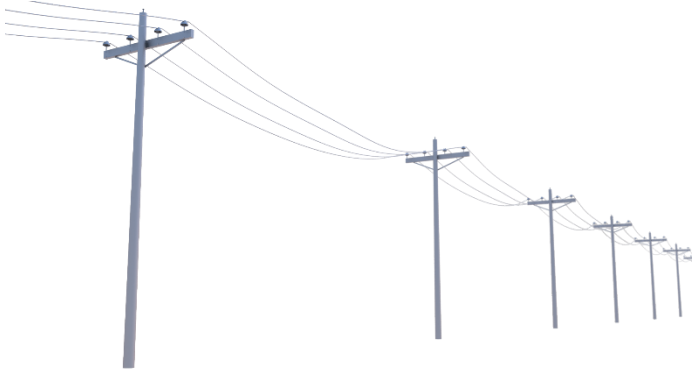
Output data

```
1 1 1 2
1 4 1 2
3 4 2 2
3 4 4 2
```

Team for C5

501	이인석	김해린
502	정성목	박은찬
503	박건희	이찬효
504	이혜림	권혁찬
505	안제현	차경민
506	이수아	홍순규
507	최시령	전영우
508	강석운	김영표
509	강동인	남진우

P5. Electricity Poles



There are n electricity poles standing on a horizontal line transmitting electricity through an electric wire. Each pole is located at a unique point between 0 and 1,000,000,000, inclusively, on the horizontal line. The intervals between two adjacent points are the same.

The office of Energy want to select k poles among the total of n poles to install electricity amplifiers for reliable energy transmission. Considering threats of electromagnetic wave interference, the k poles must be selected such that the minimum distance between two selected poles must be as large as possible.

Write a program that finds the minimum distance between two poles of the k electricity poles that will be selected by the office of Energy.

Requirements

Input

- The first line from the standard input has two numbers n and k. The first number, n stands for the number of electricity poles where $2 \leq n \leq 100{,}000$. The other number, k represents the number of poles to install amplifiers for $2 \leq k \leq n \leq 100{,}000$.
- Thereafter, n lines follow, each of which contains one number that represents x_i, the location of an electricity pole for $1 \leq x_i \leq 1{,}000{,}000{,}000$. Note that these numbers are not sorted in any order.

Output

- Print out one number to the standard output within 0.5 second.

Test case examples

Input1

```
5 3
1
10
5
7
9
```

Output1

```
4
```

Input2

```
5 4
1
10
5
7
9
```

Output2

```
2
```