1)Prince of Persia

You are given $m \times n$ dimensional array with 0 and 1. 0 means free location and 1 means obstacle. Now given two location of prince and princess you need to find out, if it's possible to go from prince to princess and if yes what is the minimum distance between them. Remember prince can only go up, down, left and right if there is no obstacle.

Example

```
0110
0101
0000
0101
```

In this 4 x 4 dimensional array suppose the prince location is (0,0) and princess is in (0,3)

Now prince can reach princess in the following path with distance 8.

0100

01**0**1

0000

0001

Sample Input format:

1. Take input the number of test case T as integer.

2. Two integer number m and n as input. Which represents the number of row and column of the array respectively

3. Take mxn numbers of input of either 1 and 0

4. location of prince and princess. You can consider the value is 0 in the location of prince and princess.

Sample Output format:

1. Print the minimum distance needed to reach princess if cannot be reached output -1

Sample Input/Output:

Sample Input	Sample Output

4
 	
3 3
100
011
010
 	
0 1
2 2
 	
9 9
000000000
000000000
000001000
000011000
000000000
000000000
000000010
000001100
000000010
0 0
8 8
 	
6 9
100000010
110001100
000000110
011000000
001100111
010100000
0 1
5 4
 	
4 4
1000
0010
0010
0001
0 0
3 3
	Case 1: -1
	Case 2: 17
	Case 3: 9
	Case 4: 7

2)

Given an N X N matrix; Write a java program that will right-rotate the matrix by 90 degrees. You can not use any additional array.

Input Format:

a) Take input the number of test case T as integer.

b) One integer number N as input which represents both the number of row and column of the square matrix.

c) Take NXN numbers as input.

Output Format:

a) Rotate the matrix and print it as output.

Sample Input/Output:

Input	Output

2
3
1 5 7
-2 3 9
4 2 0
 	
4
1 4 7 9
5 -2 8 2
-3 6 3 2
8 4 2 3
	Case -1:
	4 -2 1
	2 3 5
	0 9 7
	
	Case- 2:
	8 -3 5 1
	4 6 -2 4
	2 3 8 7
	3 2 2 9

3)

Given an *m X n* matrix; Write a java program that will right-rotate the sub-matrix (n X n, if n < m; otherwise m X m) by 90 degrees. The sub-matrix can be considered from [0,0] index. You can not use any additional array.

Input Format:

a) Take input the number of test case T as integer.

b) Two integer number m, n as input where m is the number of row and n is the number of column of the square matrix.

c) Take m X n numbers as input.

Output Format:

a) Rotate the sub-matrix and print it as output.

Sample Input/Output:

Input	Output

2
3 3
1 5 7
-2 3 9
4 2 0
 	
4 5
1 4 7 9 1
5 -2 8 2 4
-3 6 3 2 7
8 4 2 3 5
	Case -1:
	4 -2 1
	2 3 5
	0 9 7
	
	Case- 2:
	8 -3 5 1 1
	4 6 -2 4 4
	2 3 8 7 7
	3 2 2 9 5

String

1) Product Code Validation

In a superstore, every product is labeled with a product code. This product code is comprises of two parts. The first part can contain only capital letters and 6 digits. Second part is all digits and equal to the product of the first 6 digits taken in group of two from the left.

Example:

A valid product code can be as follows:

AX6BYU56UX6CV6BNT7NM   287430

This product code is valid because the first part contains only capital letters and 6 digits. And the second part is equal to the product of of the digits taken in group of two from the left. That is;

65*66*67 = 287430

Now you have to write a java program to test whether a provided product code is valid or not.

Input Format

1. Take an integer T as input which denotes number of test cases.
2. Take a string as input which denotes the complete product code where both the parts are separated by a space. The highest product code can be of size 100.

Output Format

1. For each test case print whether the provided product code is valid or not.

Sample Input/Output

Sample Input	Sample Output
4 AX6BYU56UX6CV6BNT7NM 287430 AX6BYt55JuH8ERTYH9U1OPO 343070 ERTY6YIU5GT6IO6GTR6TYR 287430 QWRE4GR5B5VD6SWE7DD8GF 196558	Case-1: Valid Case 2: Invalid Case 3: Invalid Case 4: Invalid

2)

You are given a string S. You have to find its smallest building block T of S such that by repeating the block T; k times we get S. i.e; T^k = S.

Input Format:

a. String, S.

You are preferred to use JOptionPane to take the input.

Output Format:

a. Print the smallest building block T as T^k format as mentioned above.

You are preferred to use JOptionPane to show the output.

Sample Input/Output:

Input	Output
`abababab`	`ab^4`
`abba`	`abba^1`
`abcabxabz`	`abcabxabz^1`
`aaaaa`	`a^5`
`abacabacabac`	`abac^3`
`abacabacabaca`	`abacabacabaca^1`

3)

You are given an expression which is parenthesized. You have to output whether the expression is valid or invalid. If the expression is properly parenthesized, then it is a valid expression.

Input Format:

Take a test case number T and then T number of expressions.

Sample Input/ Output:

Input	Output

3
((x+y)*(x+y*x))
((((x*(x+y)-i*j)+k)
((i-k*(x+(x*y)+j)*3)+n)
	case-1: Valid
	case-2: Invalid
	case-3: Valid

4)

You are given a string S which can contain any letter and digit. You have to separate the letters and digits into two strings.

In the output strings, each letter should be replaced such that
letter = letter + 4 (Ex: ‘a’ should be replaced by ‘e’)
and
each digit should be replaced by digit = digit +2 (Ex: ‘5’ should be replaced by ‘7’).

~~You have to wrap around the alphabet or digits for replacement. You are preferred to use JOptionPane for input and output.~~

Sample Input/ Output:

Input	Output
abcDE4215MNo75	643797 efgHIQRs
159iJk2438rsT	3714650 mNovwX