# Problem 3 – Magic Grid

You are given an **encrypted string**, a **magic number** and a **square** **matrix** **of** **numbers**. Your task is to **decrypt** the **encrypted** **message**.

In order to do that you first need to do the following:

1. You need to find the **positions** (**row** and **col**) of the **only** **two** numbers in the matrix, which **sum** is **equal to** the **magic number**.
2. You need to **sum** the **row** and **column** values in order to get the **key** for the encrypted message (a number).
3. For each **character** on an **odd position** in the string you need to **subtract** the **key value** from the character's **ASCII** **code** and for each **character** on an **even position** in the string you need to **add** the key value to the character's **ASCII** **code**. The first character of the string (at position 0) is considered even.

In the output you should print the **decrypted message**.

### Input

The input data is passed to the first JavaScript function found in your code as an **array of strings**. The **first** input line holds the **encrypted message**. The **second** input line holds the **magic number**. **Each** of the **next** **lines** holds an **array** of **numbers** separated by a **single space**. The input data will always be valid and in the format described. There is no need to check it explicitly.

### Output

### On the only output line you should print out the decrypted message.

### Constraints

* The magic number is in the range [0…999].
* The magic key is in range [1…10].
* The matrix size is in the range [2…10].
* The numbers in the matrix are separated by a single space.
* Allowed working time for your program: 0.2 seconds. Allowed memory: 16 MB.

**function** *solve*(input) {  
  
 **let** encript = input.shift();  
 **let** sum = input.shift();  
 **let** matrix = input.map(row=> row.**split**(**' '**))  
 **let** result = 0;  
  
 **for** (**let** i = 0; i < matrix.**length**; i++) {  
 **for** (**let** j = 0; j < matrix[i].length; j++) {  
 **let** current = Number(matrix[i][j]);  
  
 **for** (**let** row = 0; row < matrix.**length**; row++) {  
 **for** (**let** col = 0; col < matrix[row].length; col++) {  
 **let** current2 = Number(matrix[row][col]);  
  
 **if**((current + current2) == sum){  
 result = (Number(row) + Number(col)+ i + j);  
 **break**;  
 }  
 }  
 }  
 }  
 }  
  
 **let** word = **''**;  
 **for** (**let** i = 0; i < encript.length; i++) {  
 **if**(i%2 == 0){  
 **let** current = encript.charCodeAt(i);  
 **let** newWord = encript.charCodeAt(i) + result;  
 word += String.fromCharCode(newWord);  
 }  
 **else**{  
 **let** current = encript.charCodeAt(i);  
 **let** newWord = encript.charCodeAt(i) - result;  
 word += String.fromCharCode(newWord);  
 }  
  
 }  
 **console**.log(word);  
  
}  
  
*solve*([  
 **'EfqfNhmnkynn%`fn~'**,  
**'100'**,  
**'200 100 120 300'**,  
**'100 90 300 100'**,  
**'150 290 370 100'**,  
**'10 11 100 100'**]);

|  |  |
| --- | --- |
| **Input** | **Comments** |
| QqdvSpg  400  100 200 120  120 300 310  150 290 370 | We search for **two numbers** which sum is equal to **400**.  At position (**0**,**0**) and (**1**,**1**) are the numbers **100** and **300** which sum **meets** the magic number.  Now we **sum** the **rows** and **cols** of the **positions** of the numbers and we get **0** + **0** + **1** + **1** = **2**.  Now we **add** **2** to the **ASCII code** of each **even** symbol and we **subtract** **2** from the **ASCII code** of each **odd** symbol and we get as a result - the string **SoftUni** |
| **Output** | |
| SoftUni | |