字符匹配题目描述：

给你一个字符串数组（每个字符串均由小写字母组成）和一个字符规律（由小写字母和.和*组成），识别数组中哪些字符串可以匹配到字符规律上。

'.' 匹配任意单个字符，'*' 匹配零个或多个任意字符；判断字符串是否匹配，是要涵盖整个字符串的，而不是部分字符串。

输入描述：

第一行为空格分割的多个字符串，1<单个字符串长度<100，1<字符串个数<100

第二行为字符规律，1<=字符规律长度<=50

不需要考虑异常场景

输出描述：

匹配的字符串在数组中的下标（从0开始），多个匹配时下标升序并用,分割，若均不匹配

输出-1

补充说明：

题目描述：

给你一个字符串数组（每个字符串均由小写字母组成）和一个字符规律（由小写字母和.和*组成），识别数组中哪些字符串可以匹配到字符规律上。

'.' 匹配任意单个字符，'*' 匹配零个或多个任意字符；判断字符串是否匹配，是要涵盖整个字符串的，而不是部分字符串。

输入描述：

第一行为空格分割的多个字符串，1<单个字符串长度<100，1<字符串个数<100

第二行为字符规律，1<=字符规律长度<=50

不需要考虑异常场景

输出描述：

匹配的字符串在数组中的下标（从0开始），多个匹配时下标升序并用,分割，若均不匹配输出-1

补充说明：

```
const rl = require("readline").createInterface({ input: process.stdin });
var iter = rl[Symbol.asyncIterator]();
const readline = async () => (await iter.next()).value;


void async function () {
    // Write your code here
    let mixArr = []
    while(line = await readline()){
        mixArr.push(line)
    }
    const initArr = mixArr[0].split(' ')
    let initReg = mixArr[1]
    const regStr = initReg.replace(/\./g,'[a-z]{1}').replace(/\*/g,'[a-z]{0,}')
    const reg = new RegExp('^'+regStr+'$')
    let indexArr = []
    initArr.forEach((item,index) => {
        const result = reg.test(item)
        if(result) {
            indexArr.push(index)
```

```
        }
    })
    if(indexArr.length) {
        console.log(indexArr.join(','))
    } else {
        console.log(-1)
    }
}()
```