

1、迷宫

一天 Extense 在森林里探险的时候不小心走入了一个迷宫，迷宫可以看成是由 $n * n$ 的格点组成，每个格点只有 2 种状态，. 和 #，前者表示可以通行后者表示不能通行。同时当 Extense 处在某个格点时，他只能移动到东南西北(或者说上下左右)四个方向之一的相邻格点上，Extense 想要从点 A 走到点 B，问在不走出迷宫的情况下能不能办到。如果起点或者终点有一个不能通行(为 #)，则看成无法办到。

时间限制：3000

内存限制：65536

输入

第 1 行是测试数据的组数 k，后面跟着 k 组输入。每组测试数据的第 1 行是一个正整数 n ($1 \leq n \leq 100$)，表示迷宫的规模是 $n * n$ 的。接下来是一个 $n * n$ 的矩阵，矩阵中的元素为. 或者 #。再接下来一行是 4 个整数 ha, la, hb, lb，描述 A 处在第 ha 行, 第 la 列，B 处在第 hb 行, 第 lb 列。注意到 ha, la, hb, lb 全部是从 0 开始计数的。

输出

k 行，每行输出对应一个输入。能办到则输出“YES”，否则输出“NO”。

样例输入

```
2
3
.##
..#
#..
0 0 2 2
5
.....
###. #
..#..
###..
...#.
0 0 4 0
```

样例输出

```
YES
NO
```

2、拯救公主

多灾多难的公主又被大魔王抓走啦！国王派遣了第一勇士阿福去拯救她。

身为超级厉害的术士，同时也是阿福的好伙伴，你决定祝他一臂之力。你为阿福提供了一张大魔王根据地的地图，上面标记了阿福和公主所在的位置，以及一些不能够踏入的禁区。你还贴心地为阿福制造了一些传送门，通过一个传送门可以瞬间转移到任意一个传送门，当然阿福也可以选择不通过传送门瞬移。传送门的位置也被标记在了地图上。此外，你还查探到公主所在的地方被设下了结界，需要集齐 K 种宝石才能打开。当然，你在地图上也标记出了不同宝石所在的位置。

你希望阿福能够带着公主早日凯旋。于是在阿福出发之前，你还需要为阿福计算出他最快救出公主的时间。

地图用一个 $R \times C$ 的字符矩阵来表示。字符 S 表示阿福所在的位置，字符 E 表示公主所在的位置，字符 $\#$ 表示不能踏入的禁区，字符 $\$$ 表示传送门，字符 $.$ 表示该位置安全，数字字符 0 至 4 表示了宝石的类型。阿福每次可以从当前的位置走到他上下左右四个方向上的任意一个位置，但不能走出地图边界。阿福每走一步需要花费 1 个单位时间，从一个传送门到达另一个传送门不需要花费时间。当阿福走到宝石所在的位置时，就视为得到了该宝石，不需要花费额外时间。

时间限制：1000

内存限制：65536

输入

第一行是一个正整数 T ($1 \leq T \leq 10$)，表示一共有 T 组数据。每一组数据的第一行包含了三个用空格分开的正整数 R 、 C ($2 \leq R, C \leq 200$) 和 K ，表示地图是一个 $R \times C$ 的矩阵，而阿福需要集齐 K 种宝石才能够打开拘禁公主的结界。接下来的 R 行描述了地图的具体内容，每一行包含了 C 个字符。字符含义如题目描述中所述。保证有且仅有一个 S 和 E 。 $\$$ 的数量不超过 10 个。宝石的类型在数字 0 至 4 范围内，即不会超过 5 种宝石。

输出

对于每一组数据，输出阿福救出公主所花费的最少单位时间。若阿福无法救出公主，则输出“oop!”（只输出引号里面的内容，不输出引号）。每组数据的输出结果占一行。

样例输入

```
1
7 8 2
.....
..S..#0.
.##..1..
.0#.....
...1#...
...##E..
...1.....
```

样例输出

```
11
```

3、二叉树的深度

给定一棵二叉树，求该二叉树的深度

二叉树深度定义：从根结点到叶结点依次经过的结点（含根、叶结点）形成树的一条路径，最长路径的节点个数为树的深度

时间限制：1000

内存限制：65535

输入

第一行是一个整数 n ，表示二叉树的结点个数。二叉树结点编号从 1 到 n ，根结点为 1， $n \leq 10$ 接下来有 n 行，依次对应二叉树的 n 个节点。每行有两个整数，分别表示该

节点的左儿子和右儿子的节点编号。如果第一个（第二个）数为-1则表示没有左（右）儿子

输出

输出一个整型数，表示树的深度

样例输入

```
3
2 3
-1 -1
-1 -1
```

样例输出

```
2
```

4、Sequence

给定 m 个数字序列，每个序列包含 n 个非负整数。我们从每一个序列中选取一个数字组成一个新的序列，显然一共可以构造出 n^m 个新序列。接下来我们对每一个新的序列中的数字进行求和，一共会得到 n^m 个和，请找出最小的 n 个和

时间限制：3000

内存限制：65536

输入

输入的第一行是一个整数 T ，表示测试用例的数量，接下来是 T 个测试用例的输入 每个测试用例输入的第一行是两个正整数 m ($0 < m \leq 100$) 和 n ($0 < n \leq 2000$)，然后有 m 行，每行有 n 个数，数字之间用空格分开，表示这 m 个序列 序列中的数字不会大于 10000

输出

对每组测试用例，输出一行用空格隔开的数，表示最小的 n 个和

样例输入

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

样例输出

```
3 3 4
```