A.环形地铁

Description

有一环形地铁，一共有n*n*站，编号1-n1−*n*。

正向行驶地铁会按照1->2->3->⋯->n->1的方向行驶

反向则按照1->n->⋯->3->2->1的方向行驶。

给定相邻两站之间的地铁行驶时间（正向、反向时间相同），现在有m*m*组询问，每次询问从第x*x*站到第y*y*站的最短时间。

Input

输入只有一组数据。

第一行包含两个整数n,m*n*,*m*，分别表示地铁站数和询问次数

第二行有n*n*个整数a1,a2,…,an*a*1,*a*2,…,*an*，其中ai*ai*表示从第i*i*站正向行驶到下一站的时间。

接下来m*m*行，每行两个整数x*x*和y*y*,代表询问从第x*x*站到第y*y*站的最短时间。

(1\le n,m,ai \le 200000,1 \le x,y\le n)(1≤*n*,*m*,*ai*≤200000,1≤*x*,*y*≤*n*)

Output

输出m*m*行，第i*i*行输出第i*i*次询问的答案

Sample Input 1

5 2

1 2 3 4 5

1 3

1 5

Sample Output 1

3

5

B.三视图

Description

在X\*Y\*Z的三维空间中有N*N*个立方体，每个立方体大小为1\*1\*1，给出这N*N*个立方体的位置坐标，求出其三视图

Input

输入第一行为四个整数X,Y,Z,N*X*,*Y*,*Z*,*N*

接下来N*N*行，每行三个整数x,y,z*x*,*y*,*z*，表示其位置坐标

1 \le X,Y,Z,N \le 1000,1\le x \le X,1\le y \le Y,1\le z \le Z1≤*X*,*Y*,*Z*,*N*≤1000,1≤*x*≤*X*,1≤*y*≤*Y*,1≤*z*≤*Z*

Output

输出共Y+Z+1*Y*+*Z*+1行，前Y*Y*行每行X+Z+1*X*+*Z*+1个字符，输出正视图和左视图，两幅图之间有一列空格

接下来输出一个空行；再接下来Z*Z*行，每行X*X*个字符，表示俯视图

其中"."表示空，"x"表示有立方体

Sample Input 1

2 2 2 2

1 1 1

2 2 2

Sample Output 1

.x .x

x. x.

x.

.x

Sample Input 2

3 3 3 3

1 1 2

2 1 1

1 2 1

Sample Output 2

... ...

x.. x..

xx. xx.

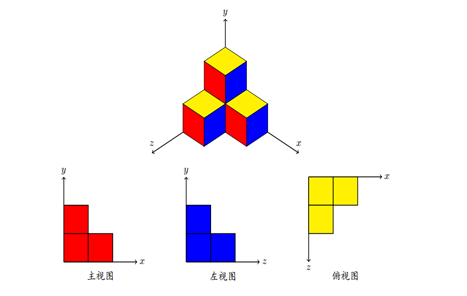
xx.

x..

...

Hint

对于样例二：



C.栈

Description

一个没有深度限制的栈的入栈序列为A1,A2,…,An*A*1,*A*2,…,*An*，A1*A*1不可以第一个出栈

求合法的出栈序列的个数。答案对998244353取模后输出。

Input

输入第一行为整数T*T*，表示有T*T*组数据(1 \le T \le 100000)(1≤*T*≤100000)

对于每组数据，输入只有一个整数n(1\le n \le 100000)*n*(1≤*n*≤100000)

Output

对于每一组数据，输出"Case x: ans "(不含引号)

x*x*表示第x*x*组测试数据，从11开始编号，ans*ans*表示答案, 答案对998244353取模后输出。

Sample Input 1

3

3

9

24

Sample Output 1

Case 1: 3

Case 2: 3432

Case 3: 508887030

D.奇偶交错排列

Description

一个1-n1−*n*的排列满足所有相邻数字奇偶性不同，那么称该排列为奇偶交错排列。

按字典序输出1-n1−*n*的所有奇偶交错排列。

Input

输入一个整数n( 2 \le n \le 11)*n*(2≤*n*≤11)

Output

输出若干行，每行一个排列。相邻数字之间以一个空格分隔，行末无空格。

请严格按照输出格式，输出不规范将直接判成Wrong answer

Sample Input 1

4

Sample Output 1

1 2 3 4

1 4 3 2

2 1 4 3

2 3 4 1

3 2 1 4

3 4 1 2

4 1 2 3

4 3 2 1

E.安全路径

Description

小花梨来到了二维世界。在n\*n*n*∗*n*的地图上

"A"表示小花梨的位置，在左下角(n,1)(*n*,1)

"D"表示小花梨要去的地方，在右上角(1,n)(1,*n*)

"."表示空白区

"#"表示石子

小花梨从左下角出发，到右上角去，每次可以往八个方向移动一格

即上、下、左、右、左上、左下、右上、右下，当然小花梨还可以原地不动。

每次做完一个移动步骤之后，所有的石子都会下落一格(在原地不动也算一个移动步骤)。

也就是说，一个石子的位置为(x,y)(*x*,*y*)，下落之后的位置是(x+1,y)(*x*+1,*y*)。

如果石子已经在最下面，那么继续下落的话就会掉出地图。

小花梨任一时刻不能与石子同处于一个位置，并且不能超出n\*n*n*∗*n*的地图之外。

请你判断小花梨能否安全到达右上角。

Input

第一行一个整数T*T*，表示有T*T*组样例(1 \le T \le 120)(1≤*T*≤120)

对于每组样例，输入第一行，为整数n(4 \le n \le 8)*n*(4≤*n*≤8)

接下来n*n*行，每行输入n*n*个字符，表示整个地图。

输入保证'A'、'D'仅出现一次，并且位置分别是(n,1),(1,n)(*n*,1),(1,*n*)

Output

对于每一组样例，先输出"Case x: "(不含引号)

x*x*表示第x*x*组测试数据，从11开始编号

如果能够安全到达输出"Yes"，否则输出"No"

Sample Input 1

2

8

.......D

........

........

........

........

.#......

#.......

A#......

8

#.####.D

.##.#.##

.....#.#

#...####

...###.#

.####.#.

.#.#.#..

A#...###

Sample Output 1

Case 1: No

Case 2: Yes

F.跳一跳

Description

给定一个长度为n*n*的数组a1,a2,…,an*a*1,*a*2,…,*an*，请判断有多少个元素可以按照下列跳跃规则跳到最后一个元素an*an*。

假设你当前位于ai*ai*，跳跃的规则是：(跳跃时从11开始计数)

如果这一步是第奇数次跳跃，可以跳到ai*ai*之后比ai*ai*大的最小的元素

如果这一步是第偶数次跳跃，可以跳到ai*ai*之后比ai*ai*小的最大的元素

如果没有满足条件的元素，则在当前ai*ai*停下来

例如对于A=[3,2,4,1,5]，从3开始，第一步从3跳到4，第二步从4跳到1，第三步从1跳到5.

Input

输入第一行为一个整数n*n*

接下来一行有n*n*个整数表示给定数组a*a*，数据保证a*a* 数组元素互不相同

(1 \le n \le 100000,1 \le ai \le 1000000)(1≤*n*≤100000,1≤*ai*≤1000000)

Output

输出一个整数，即可以跳到an*an*的个数（an*an*本身也算）

Sample Input 1

5

3 4 1 2 5

Sample Output 1

4