

《数据结构》第3次作业

题目一览

名称	代号	满分
目录树	dirtree	25
光线追踪	raytracing	25
重复单词检索	duplicity	25
亚布迪克门冲战	battle	25

习题3.1 目录树(dirtree)

【题目描述】

给出一组文件系统中的路径信息，请根据这组信息整理出覆盖它们的最小文件系统目录结构，并输出这个目录结构。路径信息中各层目录名的分隔符为“/”。

【输入】

第一行包含一个整数 n ，表示路径信息的数目。接下来 n 行为 n 个路径信息，每个路径信息的长度不会超过255个字符。目录名由英文字母、数字、“.”、“_”、“-”、“@”、“+”、“{”、“}”组成。

【输出】

目录结构。同层目录名按照ASCII码的顺序排列。如果A目录是B目录的子目录，那么B目录在A目录输出之后输出，并且在B目录输出前增加2个空格的缩进。

【样例输入】

```
6
usr/bin
usr/local/bin
bin
usr/share
usr/local/share
zip
```

【样例输出】

```
bin
usr
  bin
  local
    bin
    share
  share
zip
```

【限制】

$1 \leq n \leq 10^5$ ，整个输入的总大小不会超过30MB。

习题3.2 光线追踪(raytracing)

【题目描述】

光线追踪是图形学中场景渲染的最基本算法,其中的一个核心计算是对射线和场景中的模型求交,以确认射线遇到的第一个模型。

而模型通常都是由一系列的三角面片构成。所以,我们的任务就是设计一个程序,快速地在一系列三角面片中,计算出与指定射线相交且离射线起点最近的交点的坐标。

【输入】

第一行包含两个整数 n 和 m , n 表示三角面片的个数, m 表示要处理的射线的数目。

接下来有 n 行,分别为每个三角面片的坐标。每行有九个范围从 -10^8 到 10^8 ,并且保留两位小数的浮点数 $x_1, y_1, z_1, x_2, y_2, z_2, x_3, y_3, z_3$,表示三角面片三个顶点 $(x_1, y_1, z_1)(x_2, y_2, z_2)(x_3, y_3, z_3)$ 的三维坐标。

再接下来有 m 行,分别描述每条射线,每行有六个范围从 -10^8 到 10^8 ,并且保留两位小数的浮点数 $x_0, y_0, z_0, x_d, y_d, z_d$,表示射线的起点 (x_0, y_0, z_0) 和方向向量 (x_d, y_d, z_d) 。注意,射线方向向量未必已经规范化,其长度可能不是1,但长度不会是0。射线的起点不会在三角面片上。

【输出】

m 行,每行为对应的射线与三角面片相交且离射线起点最近的交点的坐标,保留两位小数,如果射线最终没有遇到任何一个三角面片,那么这一行输出为Miss。

【样例输入】

```
2 2
0.00 0.00 0.00 10.00 0.00 0.00 0.00 0.00 10.00
10.00 0.00 0.00 0.00 0.00 10.00 10.00 0.00 10.00
1.00 -10.00 1.00 0.00 1.00 0.00
1.00 -10.00 1.00 0.00 -1.00 0.00
```

【样例输出】

```
1.00 0.00 1.00
Miss
```

【限制】

$1 \leq n \leq 10^5, 1 \leq m \leq 10^4$ 。

习题3.3 重复单词检索(duplicity)

【题目描述】

给出一个全部为小写字母组成的单词的列表，判断里面是否有重复出现的单词，如果有，输出所有出现重复的单词，输出以单词在列表中第一次出现重复的位置为序。

【输入】

第一行包含一个整数 n ，表示列表中单词的数目。

接下来有 n 行，每行一个单词，每个单词的长度不会超过40，单词都是由英文小写字母组成。

【输出】

所有出现重复的单词。如果一个单词重复出现多次只输出一次；输出以单词第一次出现重复的位置为序。

【样例输入】

```
7
apple
banana
cake
banana
banana
apple
orange
```

【样例输出】

```
banana
apple
```

【限制】

$1 \leq n \leq 5 \times 10^5$ 。

习题3.4 亚布迪克门冲战(battle)

【题目描述】

亚布迪克门冲战, 驰援而来的亚维的幻炎舰队和统合体军的主力舰队在亚布迪克门附近的平面宇宙展开最终对决。而由于平面宇宙和通常宇宙的物理规则不一样, 双方的舰船都要依靠生成时空泡的方法来使舰船能在平面宇宙航行。相遇的时空泡可以融合成一个大的时空泡, 而且战斗也只能在时空泡内进行。

在双方的舰队还有一段距离的时候, 为了提高机动能力, 统合体军的舰队都基本上以单舰时空泡进行前进, 但是随着双方的接近, 统合体军开始通过将一些时空泡进行融合来集中部队以应付将要打响的遭遇战。

而作为亚维这边, 虽然可以监测到对方时空泡的规模 and 变化, 但是为了更好地计算敌人的状态, 需要根据敌人时空泡的合并情况记录敌舰队每个时空泡的状态, 供舰队司令随时查询, 而这个任务就交给了身为作战参谋的你。当然, 对付敌人这成千上万个时空泡, 光靠人脑是不行的, 于是你决定编写一个程序来帮忙。

程序按照指令的形式进行处理, 指令分两种, 一种是时空泡合并指令, 另一种是查询指令。 n 个时空泡根据最开始的状态已经编号(编号为0到 $n-1$)。

合并指令形式为: $M\ x\ y$, M 是表示合并指令的指令符号, 这条指令表示原来的 x 号时空泡所合并出来的时空泡(也可能还是原来的 x 号时空泡)与原来的 y 号时空泡所合并出来的时空泡(也可能还是原来的 y 号时空泡)进行了时空泡融合。

查询指令形式为: $Q\ x\ y$, Q 是表示查询指令的指令符号, 这条指令表示询问原来的 x 号时空泡和原来的 y 号时空泡现在是不是已经融合到一个时空泡里面去了。是则返回Y, 否则返回N。

【输入】

第一行为两个整数 n 和 m 。 n 表示的是开始的时候时空泡的数目。 m 表示指令的数目。

接下来的 m 行, 每行一个指令, 为 $M\ x\ y$ 的时空泡合并指令或者 $Q\ x\ y$ 的查询指令, x 和 y 的取值都在 $[0, n)$ 之间。

【输出】

对于每一条查询指令, 将对应的查询结果Y或者N输出, 每个查询一行。

【样例输入】

```
4 7
Q 0 1
M 0 1
Q 0 2
M 2 3
Q 0 1
M 1 2
Q 0 3
```

【样例输出】

```
N
N
Y
```

Y

【限制】

$1 \leq n \leq 30,000$, $1 \leq m \leq 500,000$ 。