# 1989. 罪犯人脸识别

# 题目描述

二哥很年轻的时候就成为了科学家,他的研究方向是人工智能和机器学习,最近M岛上的警察局打电话给二哥,称 有三名穷凶极恶的罪犯从M岛的监狱里越狱了,这三名罪犯的名字分别是大光头,尖嘴猴和大方脸,M岛的警方可 以提供给你一些他们三人从前的画像,他们的画像都是二值的黑白图片(即图片上的每个像素要么是1要么是0,0代 表黑色,1代表白色),照片的宽度W和高度H满足(32 pixel <= W,H <= 1000 pixel),大光头的画像上面只有一个正 圆,尖嘴猴的画像上面只有一个三角形,大方脸的画像上面只有一个正方形。监控摄像头捕捉到这三名罪犯之一的 时候会立刻通知警察,现在希望二哥能设计一种方法,告诉警察现在发现的罪犯是三人中的哪一个。

二哥最近要写论文实在太忙,就把这个任务托付给你了。人脸识别问题是机器学习的热点问题,虽然你可能没有什 么机器学习的基础知识,但是二哥觉得有了下面的一些限制,这个问题对于聪明的你来说,还是很容易的:

假定这些画像都经过了特殊处理,没有任何噪音,白色区域也不会碰到画像的边界,半径以及最长的边长一定大于 30像素。这些画像是几个才能出众的画匠在Windows下用mspaint画出来的,由于画匠们视角不同,所以罪犯的脸 有旋转、平移和尺度变换是免不了的。

画匠们还告诉二哥一个Hint(注意!这个Hint不一定是必要的,这样简单的图形不止有一种特征可以用来识别), 他们在作画的时候先确定了顶点的像素位置,然后把两个像素中心点的位置连接起来作为边界,直线穿过的像素都 涂上了黑色。至于圆形,二哥请你自己画画看。

二哥把已有的几张画像和经过转换的数据矩阵传到了网上作为你的"训练数据",供你参考,监控捕捉到的头像可能 和训练数据中的某张头像一模一样,也有可能完全不同,但是上述的所有限制仍旧满足。

### 输入格式

第一行两个整数W和H,表示罪犯画像的宽度和高度

接下来的H行,每行W个数,给出了罪犯的黑白二值画像每个像素点的值

# 输出格式

输出一行,只有一个单词

如果你的程序判断发现的罪犯是大光头,输出 circle

如果你的程序判断发现的罪犯是尖嘴猴,输出 triangle

如果你的程序判断发现的罪犯是大方脸,输出 square

#### 说明

这是一个实际的问题,不要纠结于严格的数学定义,想想怎样能让你的程序拥有更鲁棒的识别能力。

训练数据: http://acm.sjtu.edu.cn/oj/pic/1989/train.zip

时间限制: 1000ms; 内存限制: 65536kb。

### 样例输入

100 100

000000000000

000000000000000000000000000000000000
0000000000
000000000000000000000000000000000000
0000000000
000000000000000000000000000000000000
0000000000
000000000000000000000000000000000000
0000000000
000000000000000000000000000000000000000
0000000000
000000000000000000000000000000000000000
0000000000
000000000000000000000000000000000000
0000000000
000000000000000000000000000000000000
0000000000
000000000000000000000000000000000000
0000000000
000000000000000000000000000000000000
0000000000
000000000000000000000000000000000111111
0000000000
000000000000000000000000000000000111111
000000000
00000000000000000000000000000111111111
0000000000
000000000000000000000000000111111111111
0000000000
000000000000000000000000011111111111111
0000000000
000000000000000000000001111111111111111
0000000000
000000000000000000000001111111111111111
00000000000 0000000000000000000000111111
0000000000
000000000000000000000000000000000000000
00000000000
000000000000000000000000000000000000000
000000000000000000000000000000000000000
000000000000000000011111111111111111111
00000000000
000000000000000001111111111111111111111
0000000000
000000000000000011111111111111111111111
0000000000
000000000000000111111111111111111111111
0000000000
000000000000011111111111111111111111111
0000000000
000000000000011111111111111111111111111
0000000000
000000000000111111111111111111111111111
000000000

```
00000000000
00000000000
00000000000
00000000000
100000000000
100000000000
100000000000
110000000000
110000000000
110000000000
111000000000
111000000000
111000000000
111000000000
111000000000
111100000000
111100000000
111100000000
111100000000
111100000000
111100000000
111100000000
111100000000
111100000000
111100000000
111100000000
111100000000
```

```
111000000000
111000000000
111000000000
111000000000
111000000000
110000000000
110000000000
110000000000
100000000000
100000000000
100000000000
00000000000
00000000000
00000000000
00000000000
000000000000
00000000000
000000000000
000000000000
000000000000
000000000000
000000000000
000000000000
000000000000
000000000000
00000000000
```

样例输出

circle

