

Exercise 03

书7.35（猜字游戏）

（游戏：猜字游戏）编写一个猜字游戏。随机产生一个单词，提示用户一次猜测一个字母，如运行示例所示。单词中的每个字母显示为一个星号。当用户猜测正确后，正确的字母显示出来。当用户猜出一个单词，显示猜错的次数，并且询问用户是否继续对另外一个单词进行游戏。声明一个数组来存储单词，如下所示：

```
// Add any words you wish in this array  
String[] words = {"write", "that", ...};
```

```
(Guess) Enter a letter in word ***** > p [Enter]  
(Guess) Enter a letter in word p***** > r [Enter]  
(Guess) Enter a letter in word pr**r** > p [Enter]  
    p is already in the word  
(Guess) Enter a letter in word pr**r** > o [Enter]  
(Guess) Enter a letter in word pro*r** > g [Enter]  
(Guess) Enter a letter in word progr** > n [Enter]  
    n is not in the word  
(Guess) Enter a letter in word progr** > m [Enter]  
(Guess) Enter a letter in word progr*m > a [Enter]  
The word is program. You missed 1 time  
Do you want to guess another word? Enter y or n>
```

Exercise 03

书7.37（游戏：豆机）豆机，也称为梅花瓶或高尔顿瓶，他是一个用来做统计实验的设备，是用英国科学家瑟弗兰克斯高尔顿的名字来命名的。它是一个三角形状的均匀放置钉子（或钩子）的直立板子，如图7-13所示。

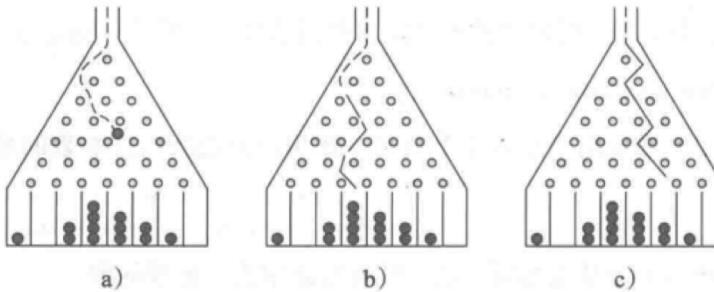


图 7-13 每个球都选取一个随机路径，然后掉入一个槽中

球都是从板子口落下的。每当球碰到钉子，它就有50%的机会落下左边或落向右边。在板子底部的槽子(slot)中都会积累一堆球。

编写程序模拟豆机。程序应该提示用户输入球的个数以及机器的槽数。打印每个球的路径模拟它的下落。例如，在图7-13b中求得路径是LLRRLLLR，而在图7-13c中球的路径是RLRRLRRL。使用条形图显示槽中秋的最终储备量。下面是程序的一个运行示例：

```
Enter the number of balls to drop: 5 [Enter]
Enter the number of slots in the bean machine: 8 [Enter]

LRLRLRRR
RRLLLRRR
LLRLLRRR
RRLLLLLL
LRLRRLRR

0
0
000
```

提示：创建一个名为 `slots` 的数组。数组 `slots` 中的每个元素存储的是一个槽中球的个数。每个球都经过一条路径落入一个槽中。路径上 R 的个数表示球落下的槽的位置。例如：对于路径 LRLRLRR 而言，球落到 `slots[4]` 中，而对路径 RRLLLLL 而言，球落到 `slots[2]` 中。

Exercise 03

书8.35（最大块）

（最大块）给定一个元素为 0 或者 1 的方阵，编写一个程序，找到一个元素都为 1 的最大的子方阵。程序提示用户输入矩阵的行数。然后显示最大的子方阵的第一个元素，以及该子方阵中的行数。下面是一个运行示例。

```
Enter the number of rows in the matrix: 5 ↵ Enter
Enter the matrix row by row:
1 0 1 0 1 ↵ Enter
1 1 1 0 1 ↵ Enter
1 0 1 1 1 ↵ Enter
1 0 1 1 1 ↵ Enter
1 0 1 1 1 ↵ Enter
The maximum square submatrix is at (2, 2) with size 3
```

程序需要实现和使用下面的方法来找到最大的子方阵：

```
public static int[] findLargestBlock(int[][] m)
```

返回值是一个包含三个值的数组。前面两个值是子方阵中的行和列的下标，第 3 个值是子方阵中的行数。