**RETOS GRADO 11**

**VIDA EN MARTE**

**Nombre Archivo Fuente:** Mensaje.java







**AGENCIAS**

 **Nombre Archivo Fuente:** Agencias.java

Una empresa automotriz tiene cinco agencias y cuenta con la información acerca de las ventas mensuales de automóviles logradas el año pasado por cada una de éstas. A partir de estos datos la empresa construyó la siguiente matriz *ventas*:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Lomas | Vallejo | Perisur | del Valle | Oriente |
| Enero |  |  |  |  |  |
| Febrero |  |  |  |  |  |
| . . . |  |  |  |  |  |
| Noviembre |  |  |  |  |  |
| Diciembre |  |  |  |  |  |

Elabore un programa modular para contestar las siguientes preguntas:

* 1. ¿Cuál fue el total de ventas en el año de la agencia Lomas?
	2. ¿Cuál fue el promedio de ventas en el mes de diciembre?
	3. ¿Qué agencia tuvo mayores ventas en el mes de mayo?
	4. ¿En qué mes se registraron las menores ventas del año, considerando todas las agencias?

Entrada

Los datos de entrada son números enteros que indican las ganancias de las 5 agencias en los 12 meses del año.

Salida.

La salida debe indicar el mensaje de la pregunta y el resultado:

Ejemplo de entrada:

100 200 150 400 300

500 200 120 130 150

200 300 160 100 600

350 800 450 700 900

600 700 130 400 820

400 420 130 600 100

500 900 400 100 200

200 700 300 750 300

100 600 120 800 400

200 400 700 900 600

300 450 500 200 350

350 650 600 300 400

Ejemplo de salida:

Total Lomas: 3800

Promedio Diciembre: 460

Agencia con mayor venta en Mes 05: Oriente con 820

Mes que registró menor venta

Lomas: 100

 Enero

 Septiembre

Vallejo: 200

 Enero

 Febrero

Perisur: 120

 Febrero

 Septiembre

Delvalle: 100

 Marzo

 Julio

Oriente: 100

 Junio

**EGIPTO**

**Nombre del archivo fuente:** Egipto.java

Hace mucho tiempo, los egipcios descubrieron que un triángulo cuyos lados miden 3, 4 y 5 tenían un ángulo recto como su ángulo más grande. Se debe determinar si otros triángulos tienen una propiedad similar.



**La entrada**

Entrada representa varios casos de prueba, seguido por una línea que contiene 0 0 0. Cada caso de prueba tiene tres números enteros positivos, menos de 30000, que indica las longitudes de los lados de un triángulo.

**La Salida**

Para cada caso de prueba, una línea que contiene "bien" si el triángulo es un triángulo rectángulo, y una línea que contiene "mal" si el triángulo no es un triángulo rectángulo.

**Ejemplo de entrada**

6 8 10

25 52 60

48 60 36

0 0 0

**Salida para entrada de muestra**

bien

mal

bien

 **UBICACIÓN DE ESTRELLAS**

**Nombre del archivo fuente:** UbicaEstrellas.java

Trabajas para los Laboratorios de Propulsión por Reacción Sputnik. En este momento es necesario que escribas un programa que lea una matriz, la cual contiene una representación digitalizada de una fotografía del cielo. Cada elemento de la matriz representa la cantidad de luz que existe en determinada región de la imagen digitalizada. El rango de intensidad va de 0 a 20. El programa permitirá localizar las regiones donde se ubica una estrella, partiendo de la siguiente información: Una estrella se encuentra en el área cubierta por el elemento i,j de la matriz si se cumple la siguiente condición:

(MD(i,j) + suma de intensidades circundantes)/5 > 10.0

Donde MD representa la matriz digitalizada.

**Entrada** La primera línea son dos enteros menores que 10 que indican el número de filas y columnas de la matriz, las siguientes líneas contienen enteros que representan las intensidades de cada posición de la matriz, separados por un blanco.

**Salida** La salida deseada es la matriz que contiene un asterisco en la posición donde está localizada una estrella, y un blanco donde no la hay. La matriz debe estar circundada por un borde que indique las coordenadas de cada estrella. La coordenada inicial es 1.

Cada elemento de la matriz debe estar separado por un espacio en blanco

**ENTRADA**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 6 | 8 |   |   |   |   |   |   |
| 0 | 3 | 4 | 20 | 15 | 0 | 6 | 8 |
| 5 | 13 | 6 | 8 | 2 | 0 | 2 | 3 |
| 2 | 6 | 2 | 2 | 3 | 0 | 10 | 0 |
| 0 | 0 | 4 | 15 | 4 | 1 | 1 | 20 |
| 0 | 0 | 7 | 2 | 6 | 9 | 10 | 4 |
| 5 | 0 | 6 | 10 | 6 | 4 | 8 | 0 |

**SALIDA**



**THE FORTIFIED FOREST**

**Nombre Archivo Fuente:** TheFortifiedForest.java

Once upon a time, in a faraway land, there lived a king. This king owned a small collection of rare and valuable trees,which had been gathered by his ancestors on their travels. To protect his trees from thieves, the king ordered that a highfence be built around them. His wizard was put in charge of the operation.

Alas, the wizard quickly noticed that the only suitable material available to build the fence was the wood from the treesthemselves. In other words, it was necessary to cut down some trees in order to build a fence around the remaining trees.

Of course, to prevent his head from being chopped off, the wizard wanted to minimize the value of the trees that had tobe cut. The wizard went to his tower and stayed there until he had found the best possible solution to the problem. Thefence was then built and everyone lived happily ever after.You are to write a program that solves the problem the wizard faced.

**INPUT**

The input contains several test cases, each of which describes a hypothetical forest. Each test case begins with a linecontaining a single integer *n*, 2 *n* 15, the number of trees in the forest. The trees are identified by consecutive integers1 to *n*. Each of the subsequent *n* lines contains 4 integers *xi, yi, vi, li* that describe a single tree. *(xi, yi)* is the position of thetree in the plane, *v*i is its value, and *li* is the length of fence that can be built using the wood of the tree. *Vi* and *li* are between 0 and 10,000.

The input ends with an empty test case *(n = 0).*

**OUTPUT**

For each test case, computer a subset of the trees such that, using the wood from that subset, the remaining trees can be enclosed in a single fence. Find the subset with minimum value. If more than one such minimum-value subset exists, choose one whit them smallest number of trees. For simplicity, regard the trees as having zero diamenter.

Display, as shown below, the test case numbers (1, 2, …), the identity of each tree to be cut, and the length of the excessfencing (accurate to two fractional digits).

Display a blank line between test cases.

**Sample Input**

6

0 0 8 3

1 4 3 2

2 1 7 1

4 1 2 3

3 5 4 6

2 3 9 8

3

3 0 10 2

5 5 20 25

7 3 30 32

0

**Output for the Sample Input**

Forest 1

Cut these trees: 2 4 5

Extra wood: 3.16

Forest 2

Cut these trees: 2

Extra wood: 15.00