Torre de cristal

Contribución de Laura Rivero y Hugo Ryckeboer

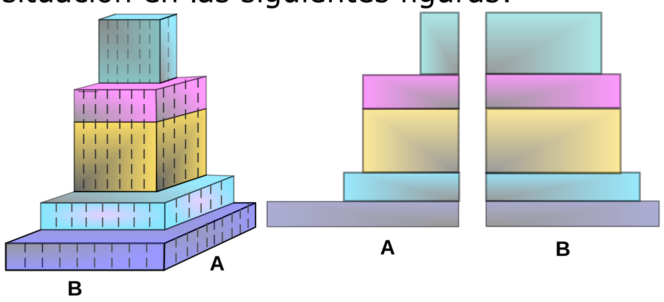
Descripción del problema

Se acerca el centenario de la fundación de una ciudad, famosa por sus establecimientos de reciclado de vidrio. La comisión encargada de los festejos ha propuesto construir una torre de cristal conmemorativa en el centro de la plaza principal, usando bloques fabricados con el vidrio que donaron los lugareños. Los bloques tendrán la forma de paralelepípedo trirectángulo.

El vidrio de los bloques no es uniforme, razón por la cual, a los efectos de asegurar la construcción, fueron pesados una vez terminados. De esta manera, se conoce el largo **L**, ancho **A**, altura **H** (medidos en metros) y peso **P** de cada bloque.

Se desea que sea la más alta que sea posible construir!!. Sin embargo existen algunas restricciones de diseño, por cuestiones estéticas, de seguridad y estabilidad de la torre:

a) No se pueden apoyar en caras laterales
b) No se puede apoyar un bloque más pesado sobre uno más liviano.
c) La base de un bloque colocado sobre otro debe apoyar íntegramente.
d) Por cuestiones de impacto estético, se desea que los bloques se alineen sobre una esquina, para que pueda lucirse sobre dos calles, lo que restringe las posibilidades de rotación de la base. Puede apreciarse la situación en las siguientes figuras:



A la izquierda la torre vista de atrás y a la derecha las vistas sobre la calles que concurren en la esquina que ocupará. Las letras A y B de la figura izquierda señalan las caras opuestas a las vistas de la derecha.

Para diseñar la torre, la comisión pide tu ayuda escribiendo la función: **torre (C, Dim, Cuales)** que, dadas las dimensiones de cada bloque y su peso determine la máxima altura que puede alcanzar la torre y su constitución indicando los números de bloques apilados, empezando por el de más abajo. Si hubiese más de una posible composición de la torre de altura máxima, cualquiera vale.

Sus parámetros son:

C: cantidad de bloques
(**1** ≤ **C** ≤ **5.000.000**)

Dim: arreglo de tamaño C de registros de **4** enteros indicando el largo **L** (**1** ≤ **L** ≤ **1.000**), el ancho **A** (**1** ≤ **A** ≤ **1.000**), la altura **H** (**1** ≤ **H** ≤ **100**) y el peso **P** de cada bloque (**1** ≤ **P** ≤ **1.000**).

Cuales: vector con los bloques usados, listados de abajo hacia arriba, seguido de un 0.

La función devuelve la altura de la torre construida.

Ejemplo

Si se presentase la siguiente situación:
Hay **C**=8 bloques de cristal; cada uno tiene las siguientes medidas (**L, A, H**) y peso **P**:

7, 5, 18 y 38
5, 6, 8 y 1
7, 5, 6 y 15
2, 8, 3 y 12
2, 6, 14 y 3
8, 6, 4 y 50
6, 4, 1 y 56
9, 10, 4 y 55

Entonces, la máxima altura de la torre es 46 metros de altura, construida con los bloques 8, 6, 1, 3 y 5.

Detalles de implementación

En un único archivo, llamado **torre.cpp**, **torre.c**, o **torre.pas**. debes enviar una función que implante la función descrita antes, usando los siguientes prototipos:

En C/C++ `long torre( long C, long Dim[][4], long Cuales[] )`

Pascal
```
TYPE DimArray = array[1..5000000] of longint ;
Function torre(C: longint ; var Dim: DimArray ;
                         var Cuales array[1..5000000] of longint )
```

Evaluador local

El evaluador local (programa para probar ejemplos propios) lee la entrada por stdin en el siguiente formato:
Línea 1: C
Línea 2+i: ($0 \leq i < C$) 4 números indicando las dimensiones de cada bloque (L, A, H) y su peso.
La función entrega el resultado por consola.

Para el caso del ejemplo la entrada sería:

```
8
7 5 18 38
5 6 8 1
7 5 6 15
2 8 3 12
2 6 14 3
8 6 4 50
6 4 1 56
9 10 4 55
```

Lo que dará por consola:--------------------

```
Altura máxima alcanzada: 46
Bloques: 8 6 1 3 5 0
```

Subtareas

Habrá casos por 25 puntos con todos los bloques de base cuadrada, y casos por 7 puntos con todos los bloques además de igual peso.

Puntaje

Se otorgarán 100 puntos por una solución completa correcta.
Habrá un puntaje parcial de 40 puntos por sólo la altura correcta.