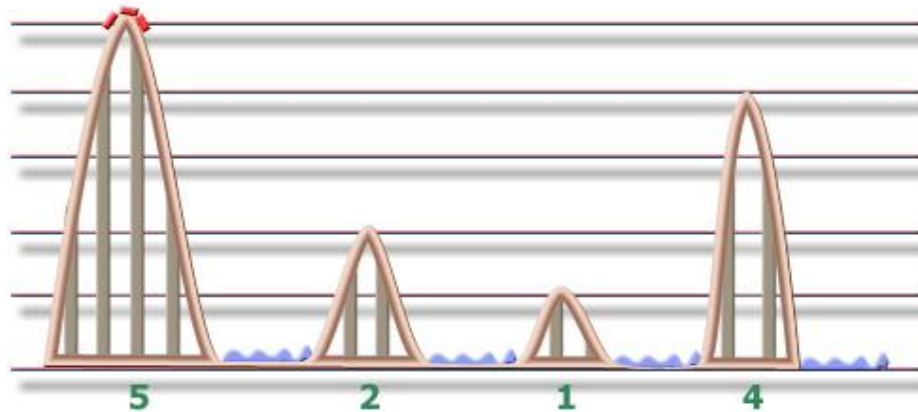


Prueba de Programación Curso 2007 2008

Problema 1.2 La Montaña Rusa

Un arquitecto está encargado de evaluar los planos de una montaña rusa que está formada por una secuencia de picos de diferentes alturas y un canal con agua a continuación de cada pico.



La figura muestra los planos de una montaña rusa cuyos picos tienen alturas 5, 2, 1 y 4 respectivamente.

Considere que un tren se deja caer desde la cima del primer pico y comienza a deslizarse hacia la derecha, gana una unidad de energía cinética por cada unidad de altura que baje (la altura de cada pico se da como un valor entero). Al llegar al canal de agua en la base pierde una unidad de energía cinética para pasar el canal y luego una unidad de energía cinética por cada unidad de altura que intente subir del siguiente pico. Si durante la subida de un pico, el tren se queda sin energía cinética (energía igual 0) antes de llegar a la cima entonces comienza a deslizarse hacia atrás en sentido contrario. Si logra llegar a la cima con energía cinética mayor o igual a cero entonces continuará con el mismo sentido de su movimiento hacia el siguiente pico.

En el ejemplo de la Figura anterior el tren parte con energía 0 del primer pico, llegará con 2 de energía al tope del segundo pico y con energía 2 al tope del tercero. Luego se quedará a mitad de la subida del cuarto pico, para empezar a bajar hacia atrás ganando 2 de energía, pasará por el canal de agua que le restará uno de energía lo que le permite subir el tercer pico nuevamente llegando a la cima con energía 0 lo que le permite seguir bajando hacia la izquierda ganando 1 de energía. Al pasar este canal se queda con energía 0 por lo que el tren no podrá intentar subir el segundo pico y se quedará detenido en el segundo canal entre el segundo y tercer picos.

Ud. debe implementar una biblioteca de clases (Class Library) para producir una dll de nombre **Prueba.dll** que contenga el espacio de nombres, la clase y el método que se describen a continuación

```
namespace Prueba {  
    class Tren{  
        public static int DondeTermina(int[] alturas)  
        {  
            ...  
        }  
    }  
}
```

El método **DondeTermina** recibe un parámetro **alturas** que es un array de enteros en que el valor de cada elemento indica la altura de cada pico (considere que todos tienen altura mayor que cero).

El método debe devolver como resultado un entero que indique el índice en el array correspondiente al pico en cuyo canal a la derecha es en el que quedaría detenido el tren (o el del último pico si el tren llega al último canal con energía mayor o igual a 0).

Para el ejemplo de la Figura se llamaría al método en la forma

```
Tren.DondeTermina(new int[] {5, 2, 1, 4})
```

y el resultado que debe devolver en este caso es **3** ya que el tren llegaría hasta el último canal.

Para ayudarlo en su trabajo usted recibirá un programa **Probador** que usando su dll dará para cada entrada la respuesta según su solución y la respuesta correcta, lo que le permitirá contrastar los resultados. Probar todas las situaciones posibles que verifiquen la corrección completa de su solución es su responsabilidad.