

[Página Principal](#) / [Mis cursos](#) / [131_15021482_ZSIFC02_MP0485_B](#) / [6. Colecciones I. Estructuras dinámicas](#)

/ [Ejercicios estructuras dinámicas en code runner](#)

Pregunta **1**

Sin finalizar

Puntúa como 1,00

Suma de dígitos con pila y aritmética decimal

<https://www.aceptaelreto.com/problem/statement.php?id=140#:~:text=Se%20trata%20de%20implementar%20un,%2B%203%20%2B%203%20%3D%2013.>

Este tipo de problema tiene cientos de soluciones. Se pide resolverlo aquí con aritmética decimal y una pila casera.

Se trata de implementar un programa que sume los dígitos de un número entero no negativo. Por ejemplo, la suma de los dígitos del 3433 es 13.

Para darle un poco más de emoción, el programa no se limitará a escribir el resultado de la suma, sino que también escribirá todos los sumandos utilizados: $3 + 4 + 3 + 3 = 13$.

Entrada

La entrada consta de una serie de casos de prueba terminados con un número negativo. Cada caso de prueba es una simple línea con un número no negativo no mayor que 109 del que habrá que sumar todos sus dígitos.

Salida

Para cada caso de prueba se mostrará una línea en la que aparecerán cada uno de los dígitos separados por el signo +, tras lo cual aparecerá el símbolo = y la suma de todos los dígitos.

Ten en cuenta que antes y después de cada símbolo (+ y =) hay un espacio.

Por ejemplo:

Entrada	Resultado
3433	3 + 4 + 3 + 3 = 13
64323	6 + 4 + 3 + 2 + 3 = 18
8	8 = 8
-1	

Respuesta: (sistema de penalización: 0 %)

1 ||

Comprobar

Pregunta 2

Sin finalizar

Puntúa como 1,00

Árbol binario que almacena expresión aritmética

Un árbol de expresión es un árbol binario en el que cada nodo no hoja se corresponde con un operador y cada nodo hoja se corresponde con un operando. Observa el siguiente el árbol de expresión



La estructura del árbol determina el orden de las operaciones, la forma del árbol produce un efecto similar a cuando escribimos paréntesis en una expresión con notación infija

Si recorres el árbol INORDEN puedes observar que almacena la siguiente expresión aritmética en notación infija.

$((9 - (5 + 2)) \times 3)$

Es una expresión totalmente parentizada, por cada operador hay siempre un paréntesis de apertura y otro de cierre. Hay paréntesis que se podrían suprimir pero para nuestro objetivo es más fácil pensar en una expresión totalmente parentizada.

El siguiente código, por el momento incompleto, obtendría del árbol anterior la expresión anterior.

```
class NodoArbol {
    NodoArbol nodoIzq;
    String datos;
    NodoArbol nodoDer;
    public NodoArbol(String datosNodo) {
        datos = datosNodo;
        nodoIzq = nodoDer = null;
    }
}
class Arbol {
    NodoArbol raiz;
}
public class App {
    public static void main(String args[]) {
        Arbol arbol = new Arbol();
        NodoArbol n=new NodoArbol("x");
        arbol.raiz=n;
        NodoArbol n1=new NodoArbol("-");
        NodoArbol n2=new NodoArbol("3");
        n.nodoIzq=n1;
        n.nodoDer=n2;
        NodoArbol n11=new NodoArbol("9");
        NodoArbol n12=new NodoArbol("+");
        n1.nodoIzq=n11;
        n1.nodoDer=n12;
        NodoArbol n121=new NodoArbol("5");
        NodoArbol n122=new NodoArbol("2");
        n12.nodoIzq=n121;
        n12.nodoDer=n122;
        //arbol.imprimir();
    }
}
```

SE PIDE: que envíes la clase Arbol y NodoArbol incorporando el método imprimir a la clase Arbol.

Por ejemplo:

Test	Resultado
------	-----------

Test	Resultado
<pre>Arbol arbol = new Arbol(); NodoArbol n=new NodoArbol("x"); arbol.raiz=n; NodoArbol n1=new NodoArbol("-"); NodoArbol n2=new NodoArbol("3"); n.nodoIzq=n1; n.nodoDer=n2; NodoArbol n11=new NodoArbol("9"); NodoArbol n12=new NodoArbol("+"); n1.nodoIzq=n11; n1.nodoDer=n12; NodoArbol n121=new NodoArbol("5"); NodoArbol n122=new NodoArbol("2"); n12.nodoIzq=n121; n12.nodoDer=n122; arbol.imprimir();</pre>	$((9-(5+2)) \times 3)$

Respuesta: (sistema de penalización: 0 %)

1 ||

Comprobar

Pregunta 3

Sin finalizar

Puntúa como 1,00

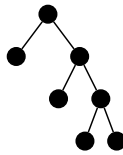
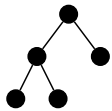
Altura de un árbol binario

<http://www.aceptaelreto.com/problem/statement.php?id=290>

La *altura* de un árbol binario se define recursivamente de la siguiente manera:

- si el árbol es vacío su altura es 0; y
- si el árbol no es vacío su altura es 1 más que el máximo de las alturas de sus hijos.

De los siguientes árboles, el de la izquierda tiene altura 3 y el de la derecha tiene altura 4.



Dado un árbol binario, ¿sabrías calcular su altura?

Entrada

La entrada comienza indicando el número de casos de prueba que vendrán a continuación. Cada caso consiste en una cadena de caracteres con la descripción de un árbol binario: el árbol vacío se representa con un punto (.); un árbol no vacío se representa con un * (que denota la raíz), seguido primero de la descripción del hijo izquierdo y después de la descripción del hijo derecho. Los árboles nunca contendrán más de 5.000 nodos.

Salida

Para cada árbol, se escribirá una línea con su altura.

Por ejemplo:

Entrada	Resultado
2	3
***.*.*..	4
**.*.*.*.*.	

Respuesta: (sistema de penalización: 0 %)

1 ||

Comprobar

Ir a...

[Examen 1º ev dic 2022 ►](#)