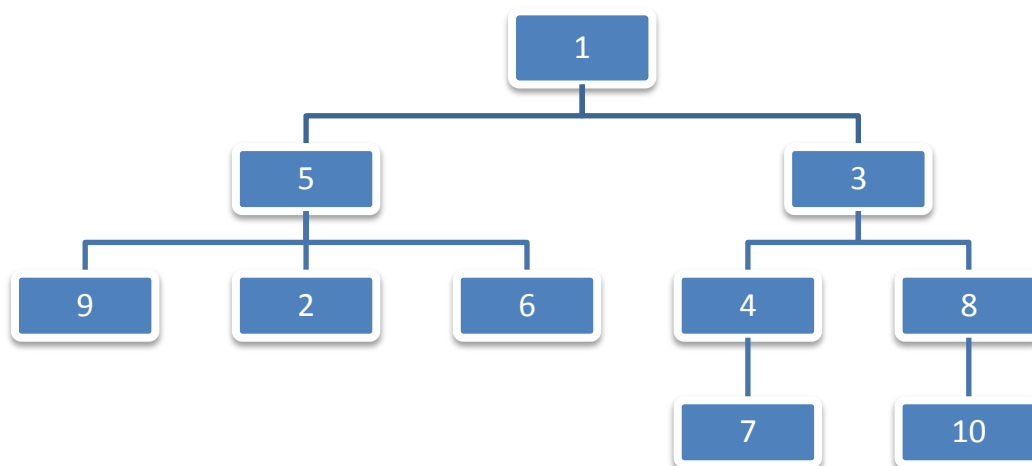


## Nodo de más descendientes

En un árbol, se dice que un nodo **M** es descendiente de orden **k** de otro nodo **N**, si **N** es ancestro de **M** en el árbol y los separan exactamente **k** niveles, por ejemplo, para el siguiente árbol:

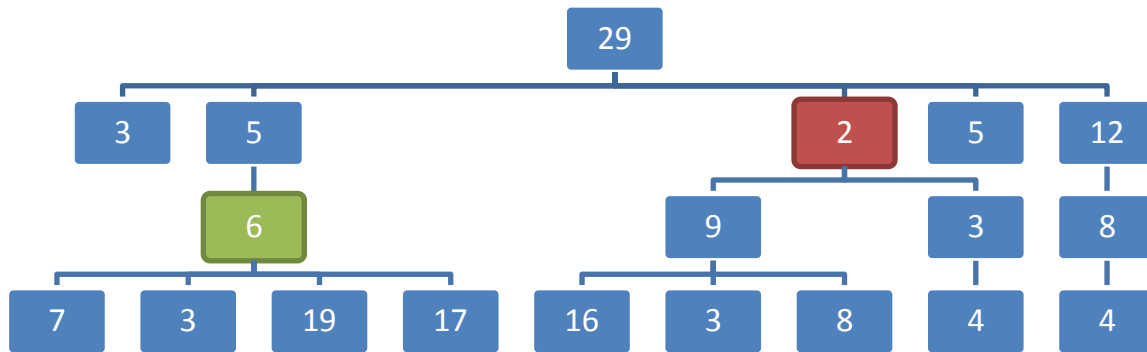


los nodos 5 y 3 son descendientes de nivel 1 del nodo raíz, los nodos del 4 al 8 son descendientes de nivel 2 del propio nodo raíz y 7 y 10 lo son de nivel 3. De igual modo el nodo 7 es descendiente de nivel 2 del nodo 3.

Dado un árbol, un entero **k** y una condición, se desea obtener aquel nodo que tenga la mayor cantidad de descendientes de nivel **k** cuyos valores cumplen con dicha condición.

Por ejemplo, en el siguiente árbol si tomamos **k=2**, y como condición que el valor del nodo sea un número par, entonces se retornaría el nodo marcado en color rojo al tener la mayor cantidad de descendientes de nivel 2 y que son valores pares (en este ejemplo 16, 8, 4). Por el contrario, si se toma **k=1** pero ahora la condición es ser número primo entonces se retorna el nodo marcado en color verde pues es el que tiene la mayor cantidad de descendientes de nivel 1 primos (7, 3, 19, 17), note que en

este ejemplo el nodo raíz también puede ser una respuesta ya que también tiene 4 descendientes de nivel 1 y que son primos (3, 5, 2, 5). En tal caso cualquiera de las dos respuestas se considerará correcta.



En la plantilla de proyecto que se le entrega, debe usted implementar el método:

```
public static Arbol<T> MasDescendientes<T>(Arbol<T> a, int k, Filtro<T> condicion)
```

El cual debe retornar el subárbol de `a` cuya raíz tenga la mayor cantidad de descendientes a nivel `k` dado para los cuales `condicion` evalúa en `true`.

Los tipos `Arbol<T>` y `Filtro<T>` están incluidos en la librería `Weboo.Utils.dll` ya adicionada como referencia en la plantilla.

Asuma que el valor que se le pase como parámetro al árbol `a` no será nunca `null` y `k` siempre será mayor que 0. Si hay más de un nodo que satisface el problema entonces pudiera darse como resultado cualquiera de estos. Tenga en cuenta que si un nodo no tiene descendientes de nivel `k` entonces para cualquiera sea la condición la cantidad de descendientes de nivel `k` que cumplen con la misma es 0.

Solo debe existir un método llamado `MasDescendientes` en su solución. Si necesita crear métodos auxiliares nómbrelos de otra manera, no sobrecargue este método.