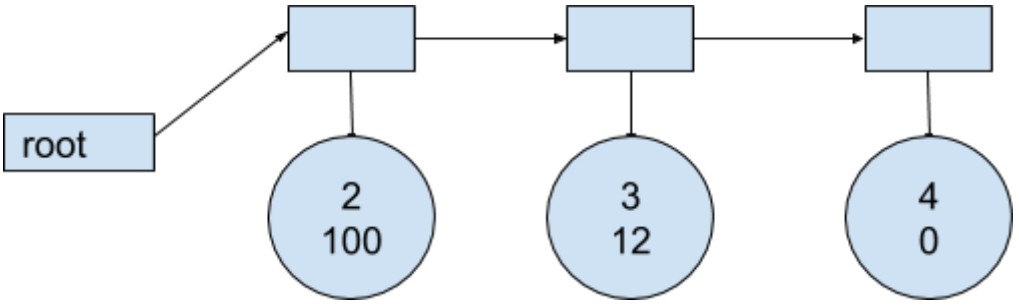
Un polinomio *p* de grado *n* y de una sola variable *x* es una función de la forma:

$$p(x) = a_n x^n + a_{n-1} x^{n-1} + a_{n-2} x^{n-2} + \ldots + a_2 x^2 + a_1 x + a_0$$

Se solicita el desarrollo de una clase llamada *Polynomial* para representar un polinomio. Para una implementación que use la memoria en forma más prolija, puede usarse una *lista simplemente vinculada*, en la que cada nodo contenga un objeto con dos atributos: el *coeficiente* del término representado, y el *exponente* de ese término (obviamente, la propia variable *x* no requiere ser representada en ninguna de las dos formas sugeridas aquí). Así, un polinomio podría quedar representado así:



Este representaría el polinomio $p(x) = 2 x^{100} + 3 x^{12} + 4$

La clase *Polynomial* debe contener los siguientes métodos:

1. *Polynomial():* este constructor debe crear un objeto *Polynomial* igual al polinomio 0(cero) (grado n = 0).
2. *Polynomial(int coef[]):* este constructor debe crear un objeto *Polynomial* cuyo grado sea igual al tamaño del arreglo coef, y cuyos coeficientes sean tomados uno a uno desde el mismo arreglo *coef* que entra como parámetro, en orden inverso, pero considerando que si algún casillero *coef[k]* es cero, entonces el término correspondiente en el polinomio no existe y *no debe agregarse*.
3. *Polynomial add(Polynomial pol):* retorna un *Polynomial* igual a la suma entre *this* y *pol*.
4. *int getCoefficient(int x):* Devuelve el valor del coeficiente del grado *x*
5. *void setCoefficient(int x, int coef):* establece el valor del coeficiente de grado x al valor *coef*
6. *float valueOf(float x):* calcula y retorna el valor del polinomio en el punto x.
7. *boolean equals(Object x):* retorna *true* si *this* es igual a *x*, y *false* en caso contrario.
8. *String toString():* retorna la representación del *Polynomial* en forma de *String*.

Generar un proyecto java con el código de la clase Polynomial. Y los casos de test para probar dicha clase.