Problem 3: Sparse polynomials

(Problem H in Mooshak POO 2015/2016)

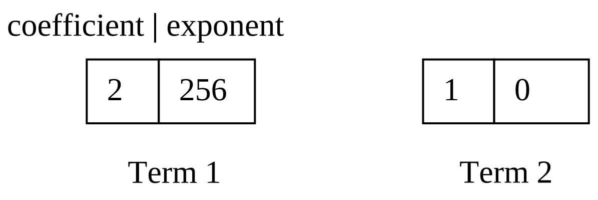
Submit:

– Your source code to mooshak http://mooshak.deei.fct.ualg.pt/~mooshak and
– Your report (within a zip file) to http://www.deei.fct.ualg.pt/POO/Entregas/

Up to November 23, 2015, 10H00

In this problem you are going to develop a program for manipulating polynomials with numeric coefficients (and exponents).

0. Read the problem description given in the appendix. Make sure you fully understand the problem. Should you have any doubt about it, do not hesitate to ask.
1. Notice that we are dealing with sparse polynomials, that is, polynomials whose number of terms with null coefficients is approximately equal to the polynomial degree.
2. For efficiency reasons it is completely out of question to use the naïf approach of employing an array to save all the polynomials coefficients (including the null coefficients). That would simplify the implementation of operations, but would be also extremely expensive in terms of the required storage space.
3. What we are going to do is that save only the terms with non-null coefficients. For instance, for the polynomial $A(x) = 2x^{256}+1$ will be save only two terms, that are roughly illustrated in the following figure:



4. Develop a class that implements a single-link list and defines its iterator.

5. Use the developed class to implement another class that implements sparse polynomials as described above. In these classes, restrict the methods only to those strictly necessary, i.e., do not code any method that is not deemed necessary for the current problem.

6. NB: Any other approach to this problem, independently of its merit, will be marked with 0 (zero) values.

7. Employ the principles and techniques of Object-Oriented Programming, specially, use nested classes and interfaces.

8. Use test-driven development.

9. Document in your report both the UML diagram(s) and the unit tests developed, as well as all the design options taken.

Appendix: Sparse Polynomials
// Leia abaixo a versão Portuguesa

Write a program that

a. sum 2 polynomials
b. multiply 2 polynomials
c. compute the value of a polynomial at a given integer point.

A polynomial is a sum of terms of the form ax^e, where:

x is the variable
a is the coefficient, and
e is the exponent

In this problem, the number of terms will never be bigger than 7 (seven). However, these terms can have very high exponents. Furthermore, variable, coefficients, and exponents are guaranteed to be integers.

Given the polynomials $P(x) = \sum_{i=0}^{n} p_i x^i$ and $Q(x) = \sum_{i=0}^{n} q_i x^i$

then $P(x) + Q(x) = \sum_{i=0}^{n} (p_i + q_i) x^i$

and $P(x).Q(x) = \sum_{i=0}^{n} (p_i x^i \sum_{j=0}^{n} q_j x^j)$

Recall also the Horner's rule which allows computing A(x) at x using a minimum of operations. This rule is based on the following observation:

$$A(x) = a_n\, x^n + a_{n-1}\, x^{n-1} + \ldots\ldots + a_1\, x + a_o = (\ldots\, (a_n\, x + a_{n-1})\, x + \ldots\ldots + a_1)\, x + a_o)$$

Input
The input has three rows.

The first row represents a polynomial with integer variable, coefficients and exponents. Furthermore, the terms of the polynomials are given by descendent order.

The second row has one (and only one) of the following possible characters:

+ (meaning that a sum of polynomials should be computed)
* (meaning that a product of polynomials should be computed)
@ (meaning that the polynomial represented in the first line should be evaluated at the integer point provided in the input next line)

The third row represents a polynomial (if the character in the previous line is either + or *) or an integer (if the character in the previous line is a @)

Polynomials are represented by a space separated sequence of integers. The first integer represents the coefficient of the term with the highest exponent. The second integer represents the exponent of the terms. The next integers have similar meaning. Terms with null coefficients are not represented. For instance, the polynomial $A(x) = 2x^{256} + 1$ is represented by the following integer sequence:

```
2 256 1 0
```

Output
The output has always one row only. This row is either one integer (whenever the required operation is @) or one polynomial (if the required operation is + or *). In this last case, the polynomial should be represented with the same format described for the input.

Sample input 1
```
1 1024 1 0
+
1 1
```

Sample output 1
```
1 1024 1 1 1 0
```

Sample input 2
```
2 100 1 10 3 1 2 0
@
1
```

Sample output 2
```
8
```

Appendix: Polinómios esparsos

// Read above the English version

Escreva um programa em Java que

a. some 2 polinómios
b. multiplique 2 polinómios
c. determine o valor do polinómio num ponto dado inteiro.

Um polinómio é uma soma de termos, em que cada termo é da forma ax^e, onde:

x é a variável
a é o coeficiente e
e é o expoente.

Neste problema o número de termos nunca será superior a 7 (sete), no entanto, esses termos podem ter expoentes elevados. Quer a variável quer os coeficientes dos polinómios são garantidamente inteiros.

Dados os polinómios $P(x) = \sum_{i=0}^{n} p_i x^i$ e $Q(x) = \sum_{i=0}^{n} q_i x^i$

então $P(x) + Q(x) = \sum_{i=0}^{n} (p_i + q_i)x^i$

e $P(x).Q(x) = \sum_{i=0}^{n} (p_i x^i \sum_{j=0}^{n} q_j x^j)$

Lembre-se ainda da regra de Horner que possibilita calcular o valor de um polinómio A(x) em x usando o menor número possível de operações. Esta regra baseia-se na seguinte observação:

$$A(x) = a_n x^n + a_{n-1} x^{n-1} + \ldots\ldots + a_1 x + a_o = (\ldots (a_n x + a_{n-1}) x + \ldots\ldots + a_1) x + a_o)$$

Input

A entrada terá sempre três linhas.
A primeira linha representará sempre um polinómio de variável, coeficientes e expoente inteiros.
A segunda linha terá um e só um dos seguintes caracteres:

+ (Indicando que se pretende efectuar a soma de polinómios)
* (Indicando que se pretende efectuar o produto de polinómios)
@ (Indicando que se pretende calcular o valor do polinómio num ponto)

A terceira linha representará um polinómio (caso o caracter da linha anterior seja + ou *) ou um inteiro (no caso desse caracter ser o @).

Os polinómios são representados por uma sequência de inteiros separados por espaços. O primeiro inteiro representa o coeficiente do termo de maior expoente, o segundo inteiro representa o expoente, e assim sucessivamente até ao último termo, de menor expoente. Não se representam termos com coeficiente nulos. Por exemplo, o polinómio $A(x) = 2x^{256} + 1$ é representado pela seguinte sequência de inteiros:

2 256 1 0

Output
Um inteiro, se a operação seleccionado for @
Um polinómio no formato descrito acima, se a operação for + ou *

Sample input 1
```
1 1024 1 0
+
1 1
```

Sample output 1
```
1 1024 1 1 1 0
```

Sample input 2
```
2 100 1 10 3 1 2 0
@
1
```

Sample output 2
```
8
```