# Prova 2 – Programação Declarativa – 2018/1
# Professor: Jânio Coutinho Canuto

## I (3,0) – Prolog

Implemente os seguintes métodos de ordenação em Prolog:

**1 (1,0).** Quicksort
**2 (1,0).** Insertion Sort
**3 (1,0).** Bubble Sort

Cada um deve ser implementado como um predicado que recebe uma lista e retorna a lista ordenada. Nomes a usar: `qSort(X,Y)`, `iSort(X,Y)`, `bSort(X,Y)` onde `Y` é a lista `X` ordenada.

## II (7,0) – Prolog ou Haskell

Implemente os seguintes métodos para manipulação de matrizes:

**1 (0,5).** Transposição
**2 (1,5).** Multiplicação
**3 (3,0).** Inversão
(tem a explicação de um método de inversão no SIGAA)

Assuma que as matrizes são listas de listas de números de ponto flutuante, as listas internas são as linhas. Ex: `[[1,2],[3,4],[5,6]]` é uma matriz com 3 linhas e 2 colunas, onde a primeira linha tem elementos 1 e 2, a segunda tem elementos 3 e 4 e a terceira tem elementos 5 e 6. Nomes a usar: `trans(A,Y)`, `mult(A,B,Y)`, `inv(A,Y)` (o último argumento só existe em Prolog, no Haskell ele é o retorno da função).

Agora implemente uma função chamada ajusta que faz os seguintes passos:

- Recebe uma lista de pontos na forma de uma matriz de $N$ linhas e 2 colunas, onde cada linha tem as coordenadas $(x,y)$ de um ponto.
- Recebe uma lista de funções escalares $[f_1, f_2, \ldots, f_k]$, ($f_i$: $a \to a$).

1. **(1,5)** Cria uma matriz `M` cuja coluna $j$ é a aplicação da função $f_j$ sobre cada uma das coordenadas $x$ dos pontos. Portanto a matriz `M` terá $N$ linhas e $k$ colunas.
2. **(0,5)** Retorna $(M'*M)^{-1}*M'*Y$, onde `M'` é `M` transposto, o expoente -1 indica a inversão e `Y` é um vetor coluna apenas com as coordenadas $y$ dos pontos. A saída deve ser uma matriz $k$ por 1.

*Curiosidade:* Essa função determina o valor dos coeficientes $c_1, c_2, \ldots, c_k$ de modo que a função $\ddot{y}_i = c_1f_1(x_i) + c_2f_2(x_i) + \ldots + c_kf_k(x_i)$ melhor represente o conjunto de dados apresentados na entrada.

Ela minimiza a soma dos erros ao quadrado, onde cada erro é a diferença entre o valor dado pela função ($\ddot{y}_i$) numa certa coordenada $x_i$ e o valor real da coordenada y neste mesmo ponto ($y_i$).

Se $f_1(x) = 1$ e $f_2(x) = x$, a função determina os coeficientes da reta que melhor aproxima os pontos, se além disso acrescentarmos $f_3(x) = x^2$, teremos a parábola que melhor representa os pontos, e assim por diante.