Primeira Prova de Compiladores - 1/2008 - 05/08/2008.

Aluno:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. (2,0) Apresente pelo menos 2 vantagens e 2 desvantagens comparando o uso de compiladores e o uso de interpretadores.
2. (2,0) Apresente através dos diagramas *tombstone* um processo de compilacao em 2 estagios  
    (**apenas desenhe o diagrama**)
3. (4,0) Mostre como seria implementado um *recursive descent parser* para a gramática abaixo, apresentando partes do seu codigo (quais os métodos e/ou classes a serem escritos, e pelo menos exemplo do código dos métodos e classes necessarios para suportar Statements e Expression).

Program ::= Statements “;” ***Programa***

Statements ::= Statement ***SimpleStmt***

| Statement “;” Statements ***MultStmt***

Statement ::= identifier ":=" Expression  ***AssignmentStmt***

Expression ::= number ***Literal***

| number "+" Expression ***AddExpression  
|*** number "-" Expression ***SubExpression***

identifier ::= “a” | “b” | “c” ***Id***

number ::= “0” | “1” | “2” | “3” | “4” | “5” | “6” | “7” | “8” | “9” ***Number***

1. (2,0) Usando a gramática acima, desenhe uma árvore sintática abstrata para o programa abaixo.

**a := 2 + 3 - 4; b := 5 + 5;**