Modularização

- Variáveis globais

Variáveis globais são declaradas na seção de declarações de variáveis do algoritmo. Elas são conhecidas e podem ser alteradas por todos os módulos que constituem o algoritmo. Quando um módulo tem uma variável local ou um parâmetro com o mesmo nome de uma variável global o módulo dará preferência à variável local ou para o parâmetro. Vamos ver alguns exemplos:

Modularização

- Variáveis globais

Exemplo:

```
algoritmo "exemplo"
var
   j:inteiro
   procedimento f1(i: inteiro)
   inicio
        j<-18
        i<-17
        escreval (i)
   fimprocedimento
inicio
   j<-3
   f1(j)
   escreva (j)
fimalgoritmo
```

Modularização

- Variáveis globais

Exemplo:

```
algoritmo "exemplo"
var
   i:inteiro
   procedimento f1(i: inteiro)
   inicio
        i<-18
        escreval (i)
   fimprocedimento
inicio
   i<-3
   f1(i)
   escreva (i)
fimalgoritmo
```

Exercício 38:

Analise o seguinte algoritmo e indique o que será impresso na saída padrão.

```
algoritmo "exercício variável global"
var num, first, sec: inteiro
    funcao func(first:inteiro; sec:inteiro):inteiro
    inicio
      first <- (first+sec)\2
      num <- num - first+1
      retorne (first)
    fimfuncao
inicio
  first <- 0
  sec <- 50
  num <- 10
  escreval ("num antes = ", num)
  escreval ("first antes = ", first)
  escreval ("sec antes = ", sec)
  num <- num + func(first, sec)
  escreval ("num depois = ", num)
  escreval ("first depois = ", first)
  escreval ("sec depois = ", sec)
fimalgoritmo
```

Interpretando como em C

10 0 50

11 0 50

No Visualg temos

10 0 50

35 0 50

Modularização

Exercício 39:

Elabore um módulo que receba um vetor com quinze elementos de valores reais e retorne o mesmo invertido. **Observação**: não é permitida a utilização de um vetor auxiliar.

```
procedimento inveter_vet(var vet: vetor [1..15] de real)
var i: inteiro
    aux: real
inicio
   para i de 1 ate 15\2 faca
      aux <- vet[i]
      vet[i] <- vet[15+1-i]
      vet[15+1-i] <- aux
   fimpara
fimprocedimento
//   O número quinze está em vermelho pois representa
//   o número de elementos do vetor.
//   A única forma de um módulo retornar um vetor é
//   através da passagem por referência.
```

Modularização

Exercício 40:

Elabore um algoritmo que manipule uma matriz de inteiros. O algoritmo deve possuir um módulo para inicializar a matriz com informações fornecidas pelo usuário, outro para apresentá-la na saída padrão, com o layout adequado, e por fim, um módulo que retorne os dois maiores valores contidos na matriz. O número de elementos contidos na matriz é fornecido pelo usuário, sendo que o número de elementos em uma dimensão não pode exceder 20. Os módulos aludidos devem ser utilizados de forma satisfatória pelo algoritmo e os módulos não devem fazer uso de variáveis globais em suas instruções.

UNIVASF