1.5 Modularização

1.5.3. exercício

Utilizando conceitos de modularização, escreva um algoritmo que leia um vetor de tamanho arbitrário com as notas de Pacotes Estatísticos, assumindo que a nota -1 é o indicativo de final de dados, e um valor inteiro entre 0 e 100 (por exemplo, 85). O algoritmo deve então calcular e escrever o percentil representado por este valor (por exemplo, 85 representa o terceiro quartil).

Supor disponíveis recursos tais como a sub-rotina ORDENA(A,B), que recebe o vetor A e o retorna ordenado em B, a função ARREDONDA(b), que retorna o inteiro proveniente do arredondamento do real b e outras que você julgar necessárias.

1.5 Modularização

1.5.3. exercício (cont.)

Sugestão:

Parta da seguinte versão inicial do algoritmo, deixando por último a definição da função PERCENTIL(DADOS, PERC).

```
algoritmo
    defina a função PERCENTIL(DADOS,PERC)
    defina os tipos das variáveis
    leia notas
    leia o percentil desejado
    calcule o percentil
    escreva o percentil
fim algoritmo
```

1.5 Modularização

1.5.3. exercício (cont.)

Solução:

A seguir, apresentamos uma solução possível para o problema, partindo da versão inicial e refinando-se primeiramente os comandos ***mais simples***.

```
ref.: leia notas
      i ← 1
      leia notas[i]
      enquanto notas[i]<>-1 faça
            i ← i+1
            leia notas[i]
      fim enquanto
fim ref.
```

```
ref.: leia o percentil desejado
      leia percent
fim ref.
```

1.5 Modularização

1.5.3. exercício (cont.)

Solução (cont.):

```
ref.: calcule o percentil
      valorPercentil ← PERCENTIL(notas,percent)
fim ref.
```

```
ref.: escreva o percentil
      escreva valorPercentil
fim ref.
```

```
ref.: defina os tipos das variáveis
      declare i, notas[1:10000],
            percent, valorPercent numérico
fim ref.
```

1.5 Modularização

1.5.3. exercício (cont.)

Solução (cont.):

Finalmente, refinando-se a função PERCENTIL, conforme apresentado ao lado.

```
ref.: defina a função PERCENTIL(DADOS,PERC)
        funcao numérico PERCENTIL(DADOS, PERC)
                {defina tipos das variáveis}
            declare tam, DADOS[:],PERC, dadosOrd[:],pos,
                 posInt, posSup, PosInf    numérico
                {ache tamanho do vetor de dados; -1 indica final do
vetor}
            tam<-0
            enquanto DADOS[tam+1]<>-1 faça
                tam ← tam+1
            fim enquanto
                {ordene dados}
            ORDENA(DADOS, dadosOrd)
                {ache posição do percentil}
            pos ← (tam+1)*( PERC /100)
                {ache percentil}
            posInt ← ARREDONDA(pos)
            se pos=posInt então {a posição já é inteira}
                PERCENTIL ← dadosOrd[posInt]
            senão {se posição não é inteira, faça uma interpolação}
                posSup ← ARREDONDA(pos+0,5)
                posInf ← ARREDONDA(pos-0,5)
                PERCENTIL <- dadosOrd[posInf] +
                        (dadosOrd[posSup]-dadosOrd[posInf])*(pos-
posInf)
            fim se
        fim funcao
fim ref.
```

1.5 Modularização

1.5.3. exercício (cont.)

Solução (final):

Inserindo-se os refinamentos no algoritmo, tem-se ao lado o algoritmo completo.

```
algoritmo
        {defina a função PERCENTIL(DADOS,PERC)}
    funcao numérico PERCENTIL(DADOS, PERC)
        {defina tipos das variáveis}
        declare     tam, DADOS[:], PERC, dadosOrd[:], pos,
                    posInt, posSup, posInf  numérico
        {ache tamanho do vetor de dados; -1 indica final do vetor}
        tam<-0
        enquanto DADOS[tam+1]<>-1 faça
            tam <- tam+1
        fim enquanto
            {ordene dados}
        ORDENA(DADOS,dadosOrd)
            {ache posição do percentil}
        pos <- (tam+1)*( PERC /100)
            {ache percentil}
        posInt <- ARREDONDA(pos)
        se pos=posInt então {a posição já é inteira}
            PERCENTIL <- dadosOrd[posInt]
        senão {se posição não é inteira, faça uma interpolação}
            posSup <- ARREDONDA(pos+0,5)
            posInf <- ARREDONDA(pos-0,5)
            PERCENTIL <- dadosOrd[posInf] +
                (dadosOrd[posSup]-dadosOrd[posInf])*(pos-posInf)
        fim se
    fim funcao
        {defina os tipos das variáveis}
    declare i, notas[1:10000], percent, valorPercent numérico
        {leia notas}
    i<-1
    leia notas[i]
    enquanto notas[i]<>-1 faça
        i <- i+1
        leia notas[i]
    fim enquanto
        {leia o percentil desejado}
    leia percent
        {calcule o percentil}
    valorPercentil <- PERCENTIL(notas, percent)
        {escreva o percentil}
    escreva valorPercentil
fim algoritmo
```