**Avaliação – Laboratório de Programação**

Prof. Msc. Lucas Gonçalves Nadalete

Nome: \_Fábio Lucas Romeiro de Castro\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Data: \_30\_/ \_04\_/ \_2018\_

Semana 25/03 à 31/03 – Exercícios 30 à 35

**Elaborar**: Algoritmo elaborado em Pseudocódigo (ver Refcard de Pseudocódigo disponível)

**Capítulo 3 – Exercício 30** – Calcule o imposto de renda de um grupo de dez contribuintes, considerando que os dados de cada contribuinte, número do CPF, número de dependentes e renda mensal são valores fornecidos pelo usuário. Para cada contribuinte será feito um desconto de 5% do salário mínimo por dependente.

Os valores da alíquota para o cálculo do imposto são:

|  |  |
| --- | --- |
| Renda liquida | Alíquota |
| Até 2 salários mínimos | Isento |
| 2 a 3 salários mínimos | 5% |
| 3 a 5 salários mínimos | 10% |
| 5 a 7 salários mínimos | 15% |
| Acima de 7 salários mínimos | 20% |

Observe que deve ser fornecido o valor atual do salário mínimo para que o algoritmo calcule os valores corretamente.

Resolução:

Algoritmo imposto\_de\_renda

var   
 cpf :Caracter  
 i, numDependentes, numeroSalariosMinimos :Inteiro  
 rendaMensal, imposto :Real

const   
 valorSalMinimo <- 954

inicio

para i <- 1 até 10 passo 1 faça  
 escreva "Digite seu CPF:"  
 leia cpf  
 escreva "Digite o numero de dependentes: "  
 leia numDependentes  
 escreva "Digite sua renda mensal: "  
 leia rendaMensal

numeroSalariosMinimos <- rendaMensal / valorSalMinimo

se (numeroSalariosMinimos <= 2) então  
 imposto <- 0  
 fim se

se (numeroSalariosMinimos <= 3 e numeroSalariosMinimos > 2) então  
 imposto <- rendaMensal \* (5/100)  
 fim se

se (numeroSalariosMinimos <= 5 e numeroSalariosMinimos > 3) então  
 imposto <- rendaMensal \* (10/100)  
 fim se

se (numeroSalariosMinimos <= 7 e numeroSalariosMinimos > 5) então  
 imposto <- rendaMensal \* (15/100)  
 fim se

se (numeroSalariosMinimos > 7) então  
 imposto <- rendaMensal \* (20/100)  
 fim se

imposto <- imposto – (numDependentes\*(rendaMensal\*(5/100)))

se (imposto > 0) então  
 escreva "Senhor(a) portador(a) do CPF: ", cpf  
 escreva "O valor do seu imposto de renda é de R$", imposto  
 fim se  
 fim para

fim

**Capítulo 3 – Exercício 31** –Foi realizada uma pesquisa sobre algumas características físicas da população de uma certa região, a qual coletou os seguintes dados referentes a cada habitante para análise:

* Sexo (‘M’ – masculino ou ‘F’ - feminino);
* Cor dos olhos (‘A’ – azuis, ‘V’ – verdes ou ‘C’ - castanhos);
* Cor dos cabelos (‘L’ – loiros, ‘C’ – castanhos ou ‘P’ - pretos);
* Idade.

Faça um algoritmo que determine e escreva:

* A maior idade dos habitantes;
* A porcentagem entre os indivíduos do sexo masculino, cujo idade está entre 18 e 35 anos, inclusive, e que tenham olhos verdes e cabelos loiros.

O final do conjunto de habitantes é reconhecido pelo valor -1 entrando como idade.

Resolução:

Algoritmo caracteristicas\_fisicas

var   
 sexo, codDosOlhos, codDoCabelo :Caracter  
 numHabitantes, idade, maiorIdade, qtdM, qtdF, idade :Inteiro   
 totalPessoas, qtdMVL :Inteiro  
 porcMVL :Real

inicio

maiorIdade <-0  
 qtdM <-0  
 qtdF <-0  
 qtdMVL <-0  
 idade <- 1

enquanto (idade != -1) faça  
 escreva "Digite sua idade:"  
 leia idade

se (idade != -1) então  
 se (idade > maiorIdade) então  
 maiorIdade <- idade  
 fim se

escreva "Qual seu sexo? (M/F)"  
 leia sexo

se (sexo == 'M') então  
 qtdM <- qtdM + 1  
 fim se

se (sexo == 'F') então  
 qtdF <- qtdF + 1  
 fim se

escreva "Qual a cor dos seus olhos? (A = Azuis / V = Verdes / C = Castanhos)"  
 leia codDosOlhos

escreva "Qual a cor do seu cabelo? (L = Loiro / C = Castanhos / P = Pretos)"  
 leia codDoCabelo

totalPessoas <- qtdF + qtdM

se (sexo == 'M' e idade > 18 e idade < 35 e corDosOlhos == 'V' e   
 corDoCabelo == 'L') então  
 qtdMVL <- qtdMVL + 1  
 fim se  
 fim se  
 fim enquanto

escreva "A maior idade dos habitantes é de ", maiorIdade

porcMVL <- (qtdMVL \* 100) / totalPessoas

escreva "A porcentagem de indivíduos do sexo masculino com idade entre 18 e 35,  
 com olhos verdes e cabelos loiros é de ",porcMVL, "%"

fim

**Capítulo 3 – Exercício 32** – Anacleto tem 1,50 metro e cresce 2 centímetros por ano, enquanto Felisberto tem 1,10 metro e cresce 3 centímetros por ano. Construa um algoritmo que calcule e imprima quantos anos serão necessários para que Felisberto seja maior que Anacleto.

Resolução:

Algoritmo Felisberto\_vai\_ficar\_maior

var tamanhoAnacleto, tamanhoFelisberto :Real  
 anos :Inteiro

inicio

tamanhoAnacleto <- 1.5  
 tamanhoFelisberto <- 1.1  
 anos <- 0

enquanto (tamanhoAnacleto > tamanhoFelisberto) faça  
 tamanhoAnacleto <- tamanhoAnacleto + 0.2  
 tamanhoFelisberto <- tamanhoFelisberto + 0.3  
 anos <- anos + 1  
 fim enquanto

escreva "Serão necessários ", anos, " anos para que Felisberto fique maior do que

Anacleto"

fim

**Capítulo 3 – Exercício 33** – Realizou-se uma pesquisa para determinar alguns dados estatísticos em relação ao conjunto de crianças nascidas em um certo período de uma determinada maternidade. Construa um algoritmo que leia o número de crianças nascidas nesse período e, depois, em um numero indeterminado de vezes, o sexo de um recém-nascido prematuro (‘M’ – masculino ou ‘F’ - feminino) e o numero de dias que este foi mantido na incubadora.

Como finalizador, teremos a letra ‘X’ no lugar do sexo da criança.

Determine e imprima:

* A porcentagem de recém-nascidos prematuros;
* A porcentagem de recém-nascidos meninos e meninas do total de prematuros;
* A media de dias de permanência dos recém-nascidos prematuros na incubadora;
* O maior numero de dias que um recém-nascido prematuro permaneceu na incubadora;

Resolução:

Algoritmo prematuros

var porcPrematuro,porcMPre, porcFPre, mediaDeDias :Real  
 sexo, prematuro :Caractere  
 qtdPrematuro, diasDeIncubadora, totalDeCriancas, qtdMPre, qtdFPre,  
 somaDeDias, maiorNumDias :Inteiro

inicio

sexo <- “”  
 qtdPrematuro <- 0  
 totalDeCriancas <- 0  
 maiorNumDias <- 0  
 somaDeDias <- 0  
 qtdMPre <- 0  
 qtdFPre <- 0

enquanto (sexo <> 'X') faça  
 escreva "Digite o sexo da criança (M/F):"  
 leia sexo

se(sexo <> X) então  
 totalDeCriancas <- totalDeCriancas + 1  
 escreva "Seu filho foi prematuro? (S/N)"  
 leia prematuro

se(prematuro == 'S') então  
 qtdPrematuro <- qtdPrematuro + 1

se(sexo == 'M') então  
 qtdMPre <- qtdMPre + 1  
 fim se

se(sexo == 'F') então  
 qtdFPre <- qtdFPre + 1  
 fim se  
 fim se

escreva "Quantos dias ele ficou na incubadora?"  
 leia diasDeIncubadora

somaDeDias <- somaDeDias + diasDeIncubadora

se(diasDeIncubadoras > maiorNumDias) então  
 maiorNumDias <- diasDeIncubadora  
 fim se  
 fim se  
 fim enquanto

porcPrematuro <- (qtdPrematuro \* 100) / totalDeCriancas  
 porcMPre <- (qtdMPre \* 100)/totalDeCriancas  
 porcFPre <- (qtdFPre \* 100)/totalDeCriancas  
 mediaDeDias <- somaDeDias / qtdPrematuro

escreva "A porcentagem de recém-nascidos prematuros é de: ", porcPrematuro, "%"  
 escreva "A porcentagem de recem-nascidos meninos do total de prematuros é de ",  
 porcMPre, "%"  
 escreva "A porcentagem de recem-nascidos meninas do total de prematuros é de ",  
 porcFPre, "%"  
 escreva "A media de dias de recem nascidos prematuros na incubadora é de ",   
 mediaDeDias, " dias"  
 escreva "O maior numero de dias que um recem nascido prematuro permaneceu na  
 incubadoa foi de ," maiorNumDias, " dias"

fim

**Capítulo 3 – Exercício 34** – Um cinema possui capacidade de 100 lugares e está sempre com ocupação total. Certo dia cada espectador respondeu a um questionário, no qual constava:

* Sua idade;
* Sua opinião em relação ao filme, segundo as seguintes notas:

|  |  |
| --- | --- |
| Nota | Significado |
| A | Ótimo |
| B | Bom |
| C | Regular |
| D | Ruim |
| E | Péssimo |

Elabore um algoritmo que, lendo esses dados, calcule e imprima:

* A quantidade de respostas Ótimo;
* A diferença porcentual entre respostas Bom e Regular;
* A media de idade das pessoas que responderam Ruim;
* A porcentagem de resposta Péssimo e a maior idade que utilizou essa opção;
* A diferença de idade entre a maior idade que respondeu Ótimo e a maior idade que respondeu Ruim;

Resolução:

Algoritmo classificacao\_cinema

var idade, somaIdade, qtdOtimo, qtdBom, qtdRegular, qtdRuim, qtdPessimo, total,  
 maiorIdadePessimo, maiorIdadeOtimo, maiorIdadeRuim :Inteiro  
 opniao :Caractere  
 porcRegular, porcBom, mediaIdade, porcPessimo :Real

inicio

somaIdade <- 0  
 qtdOtimo <- 0  
 qtdBom <- 0  
 qtdRegular <- 0   
 qtdRuim <- 0  
 qtdPessimo <- 0  
 maiorIdadePessimo <- 0  
 maiorIdadeOtimo <- 0  
 maiorIdadeRuim <- 0

para numDeEspectadores <- 1 até 100 passo 1 faça  
 escreva "Digite sua idade"  
 leia idade  
 escreva "Qual sua opnião sobre o filme? (A = Otimo / B = Bom / C = Regular /   
 D = Ruim / E = Pessimo)"  
 leia opniao

se (opniao == 'A') então  
 qtdOtimo <- qtdOtimo + 1

se (idade > maiorIdadeOtimo) então  
 maiorIdadeOtimo <- idade  
 fim se  
 fim se

se (opniao == 'B') então  
 qtdBom <- qtdBom + 1  
 fim se

se (opniao == 'C') então  
 qtdRegular <- qtdRegular + 1  
 fim se

se (opniao == 'D') então  
 qtdRuim <- qtdRuim + 1  
 somaIdade <- somaIdade + idade

se (idade > maiorIdadeRuim) então  
 maiorIdadeRuim <- idade  
 fim se  
 fim se

se (opniao == 'E') então  
 qtdPessimo <- qtdPessimo + 1

se (idade > maiorIdadePessimo) então  
 maiorIdadePessimo <- idade  
 fim se  
 fim se  
 fim para

total <- qtdOtimo + qtdBom + qtdRegular + qtdRuim + qtdPessimo  
 porcBom <- (qtdBom \* 100) / total  
 porcRegular <- (qtdRegular \* 100)/total  
 porcPessimo <- (qtdPessimo \* 100) / total  
 mediaIdade <- somaIdade / qtdRuim

escreva "A quantidade de respostas ótimas foi de ", qtdOtimo  
 escreva "A diferença de porcentual entre respostas Bom e Regular é de ",   
 porcBom - porcRegular, "%"  
 escreva "A media de idade das pessoas que responderam Ruim foi de: ",   
 mediaIdade, " anos"  
 escreva "A porcentagem de respostas pessimas é de : ", porcPessimo, "%"

escreva "A pessoa com maior idade que votou pessimo tem ",   
 maiorIdadePessimo, " anos"  
 escreva "A diferença de idade entre a maior idade que respondeu Otimo e a   
 maior idade que respondeu Ruim é de ",   
 maiorIdadeOtimo - maiorIdadeRuim, " anos"  
fim

**Capítulo 3 – Exercício 35** – Em um prédio há três elevadores denominados A, B e C. Para otimizar o sistema de controle dos elevadores foi realizado um levantamento no qual cada usuário respondia:

* O elevador que utilizava com mais frequência;
* O período em que utilizava o elevador, entre
  + ‘M’ – matutino;
  + ‘V’ = vespertino;
  + ‘N’ = noturno.

Construa um algoritmo que calcule e imprima:

* Qual é o elevador mais frequentado e em que o período, se concentra o maior fluxo;
* Qual o período mais usado de todos e a que elevador pertence;
* Qual a diferença porcentual entre o mais usado dos horários e o menos usado;
* Qual porcentagem sobre o total de serviços prestados do elevador de media utilização.

Resolução:

Algoritmo elevadores

var  
 elevador, periodo :Caracter  
 a, b, c, m, v, n, elevadorMaisUsado, periodoMaisUsado, i :Inteiro  
 porcElevadorMaisUsado :Real

inicio

a <- 0  
 b <- 0  
 c <- 0  
 m <- 0  
 v <- 0  
 n <- 0  
 elevadorMaisUsado <- 0  
 periodoMaisUsado <- 0  
 i <- 0

escreva "Qual o elevador mais utilizado por voce? (A / B / C)"  
 leia elevador

escreva "Qual o periodo em que voce mais o utiliza? (M = Matutino / V = Vespertino  
 / N = Noturno)"  
 leia periodo

enquanto (elevador <> '') faça

se(elevador == 'a') então  
 a <- a + 1  
 fim se

se(elevador == 'b') então  
 b <- b + 1  
 fim se

se(elevador == 'c') então  
 c <- c + 1  
 fim se

se(periodo == 'm') então  
 m <- m + 1  
 fim se

se(periodo == 'v') então  
 v <- v + 1  
 fim se

se(periodo == 'n') então  
 n <- n + 1  
 fim se

i <- i + 1  
 fim enquanto

se(a>b e b>c) então  
 elevadorMaisUsado <- 'A'  
 escreva "O elevador A tem o maior fluxo de pessoas"  
 fim se

se(b>a e a>c) então  
 elevadorMaisUsado <- 'B'  
 escreva "O elevador B tem o maior fluxo de pessoas"  
 fim se

se(c>a e a>b) então  
 elevadorMaisUsado <- 'C'  
 escreva "O elevador B tem o maior fluxo de pessoas"  
 fim se

se(m>v e v>n) então  
 escreva "Matutino é o perído com maior fluxo de pessoas"  
 fim se

se(v>m e m>n) então  
 escreva "Vespertino é o perído com maior fluxo de pessoas"  
 fim se

se(n>m e n>v) então  
 escreva "Noturno é o perído com maior fluxo de pessoas"  
 fim se

porcElevadorMaisUsado <- (elevadorMaisUsado \* 100) / i

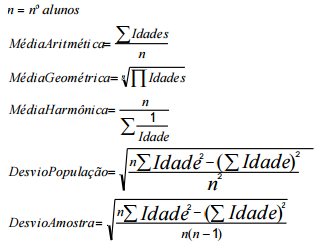
escreva "A porcentagem do elevador mais utilizado é de ", porcElevadorMaisUsado,  
 "%"

fim

Exercícios – Estruturas de Repetição

**Elaborar**: Algoritmo em Pseudocódigo (ver Refcard de Pseudocódigo disponível)

Ler do teclado a idade dos alunos de uma turma (N alunos). Ao final, quando se digita idade igual a zero, calcular a média aritmética das idades, a média geométrica, a média harmônica e o desvio padrão (população e amostral). Imprimir as médias e o desvio-padrão calculados.



Algoritmo idade\_dos\_alunos

var

n, soma, somaMH, multiMG, somaDP :Inteiro  
 idade, MG, MA, MH, DP, DA :Real

inicio

idade <- 1  
 n <- 0  
 soma <- 0  
 multiMG <- 1  
 somaMH <- 0  
 enquanto (idade <> 0) faça

escreva (“Digite a idade do aluno (ou digite 0 para sair):”)  
 leia (idade)  
 somaMA <- somaMA + idade  
 multiMG <- multiMG \* idade  
 somaMH <- somaMH + (1/idade)  
 somaDP <- somaDP + (idade\*\*2)   
 n <- n + 1

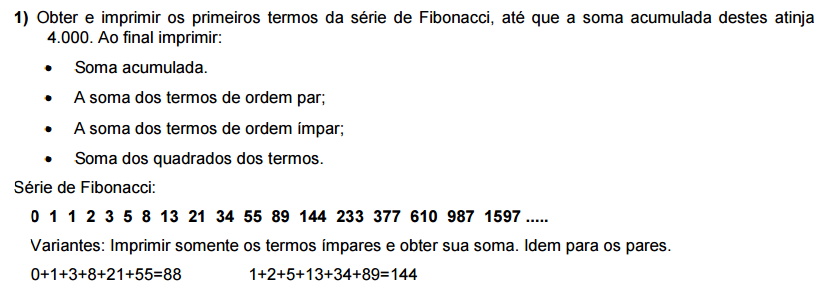
fim-enquanto  
 n <- n - 1

MA <- soma/n  
 MG <- rad(multiMG, n)  
 MH <- n/somaMH  
 DP <- rad(((somaDP – (soma)\*\*2)/n\*\*2),2)  
 DA <- rad(((somaDP – (soma)\*\*2)/n\*(n-1)),2)  
 escreva (“Média Aritmética = ”, MA)  
 escreva (“Média Geométrica = ”, MG)  
 escreva (“Média Harmônica = ”, MH)  
 escreva (“Desvio População = ”, DP)  
 escreva (“Desvio Amostra= ”, DA)

fim

Exercícios – Estruturas de Repetição

**Elaborar**: Algoritmo em Pseudocódigo (ver Refcard de Pseudocódigo disponível)



Algoritmo soma\_fibonacci

var somaAcumulada, somaPar, somaImpar, somaQuad, a, b:Inteiro  
inicio   
 a <- 0  
 b <- 1  
 somaImpar <- 0  
 somaPar <- 0  
 somaQuad <- 0  
 somaAcumulada <- 0

enquanto (soma <= 4000) faça  
 somaAcumulada <- somaAcumulada + b  
 se (MOD(b,2) == 0) então  
 somaPar <- somaPar + b  
 senão  
 somaImpar <- somaImpar + b  
 fim-se  
 somaQuad <- somaQuad + (b\*\*2)  
 a <- b  
 b <- a + b  
 fim-enquanto  
 escreva (“Soma dos valores acumulados = ”, somaAcumulada)  
 escreva (“Soma dos números pares = ”, somaPar)  
 escreva (“Soma dos números impares = ”, somaImpar)  
 escreva (“Soma dos quadrados dos termos = ”, somaQuad)

fim

Semana 08/04 à 11/04 – Exercícios 6 à 9 (Estrutura de Dados Homogênea Unidimensionais)

**Elaborar**: Algoritmo elaborado em Pseudocódigo (ver Refcard de Pseudocódigo disponível)

**Capítulo 4 – Exercício 6** – Construa um algoritmo que permita informar dados para 2 vetores inteiros de 20 posições, e apresente a intersecção dos vetores. Lembrando que intersecção são os elementos repetidos em ambos os vetores, mas sem repetição (Cada número pode aparecer uma única vez no resultado):

Algoritmo interseccao

tipo vinteiros = vetor [1..20] de inteiros

var   
 vinteiros: conjuntoA, conjuntoB, intersecção  
 contador :Inteiro  
 existente :Logico

inicio

para i<-1 até 20 passo 1 faça  
 escreva "Digite um elemento do conjunto A"  
 leia conjuntoA[i]  
 fim para

para i<-1 até 20 passo 1 faça  
 escreva "Digite um elemento do conjunto B"  
 leia conjuntoB[i]  
 fim para

contador<-0

para i<-1 até 20 passo 1 faça

para x<-1 até 20 passo 1 faça  
 se (conjuntoA[i] == conjuntoB[x]) então  
 existente<- falso

para y<-1 até 20 passo 1 faça

se(conjuntoA[i] == interseccao[y]) então  
 existente <- verdadeiro  
 fim se  
 fim para

se(existente == falso) então  
 contador <- contador + 1  
 interseccao[i] <- conjuntoA[i]  
 fim se  
 fim se  
 fim para  
 fim para

para y<-1 até contador passo 1 faça  
 escreva interseccao[y]  
 fim para   
fim

**Capítulo 4 – Exercício 7** – Construa um algoritmo que permita informar dados para 2 vetores inteiros de 20 posições e apresente o conjunto união dos vetores. Lembrando que conjunto união são todos os elementos que existem em ambos os vetores, mas sem repetição (Cada número pode aparecer uma única vez no resultado).

Algoritmo uniao

tipo vinteiros = vetor [1..20] de inteiros

var   
 vinteiros: conjuntoA, conjuntoB, união  
 contador :Inteiro  
 acabado :Logico

inicio

para i<-1 até 20 faça  
 escreva "Digite um elemento do conjunto A"  
 leia conjuntoA[i]

escreva "Digite um elemento do conjunto B"  
 leia conjuntoB[i]  
 fim para

contador<-1  
 contador2<-1  
 contador3<-1

enquanto (acabado == falso) faça

se conjuntoA[i] == conjuntoB[i] então  
 uniao[contador2] <-conjuntoA[contador]  
 contador2 <- contador2 + 1  
 fim se

contador <- contador + 1

se(contador > 20) então  
 contador <- 1  
 contador3<- contador3 + 1  
 fim se

se(contador3 > 20) então  
 acabado <- verdadeiro  
 fim se  
 fim enquanto

contador2 <- 1  
 contador3 <- 20  
 acabado <- falso

enquanto (terminado == falso) faça

para i<- 1 até contador3 passo 1 faça

se (uniao[contador2+i] == uniao[contador2]) então  
 uniao[contador2+i] <- 0  
 fim se

se(i == contador3) então  
 contador2 <- contador2 + 1  
 contador3 <- contador3 – 1  
 fim se

se(contador2 > 20) então  
 acabado <- verdadeiro  
 fim se  
 fim para  
 fim enquanto

escreva uniao

fim

**Capítulo 4 – Exercício 8** – Crie um algoritmo que leia a pontuação final de 200 provas de um concurso e os nomes dos respectivos participantes, e apresente um ranking dos colocados que obtiveram mais de 70 pontos.

Algoritmo ranking

tipo vreais = vetor [1..200] de reais

tipo vcaracter = vetor [1..200] de caracter

var   
 vreais: pontuacao  
 vcaracter :participante, ranking

inicio

para i<-1 até 200 passo 1 faça  
 escreva "Digite seu nome: "  
 leia participante[i]

escreva "Digite sua pontuacao: "  
 leia pontuacao[i]

se (pontuacao[i] > 70) então  
 ranking[i]<- "Nome: ", participante[i], " | Pontuacao: ", pontuacao[i], " pontos"   
 escreve ranking[i]  
 fim se  
 fim para

fim

**Capítulo 4 – Exercício 9** – Dado um vetor com dados de 50 alturas, elabore um algoritmo que permita calcular:

1. A média das alturas;
2. O desvio padrão das alturas. Lembrando que desvio padrão é dado por (∑(alturas²)/número de alturas) – media²;
3. A moda das alturas. Lembrando que moda é o valor que tem maior incidência de repetições;
4. A mediana das alturas. Lembrando que a mediana é o elemento central de uma lista ordenada;

Algoritmo alturas

tipo   
 vreais = vetor [1..50] de reais  
 vinteiros = vetor [1..50] de inteiros

var   
 vreais: alturas  
 vinteiro: contador  
 soma, media, desvioPadrao :Reais  
 maisRepitido :Inteiro

inicio

soma<-0  
 maisRepitido<-0

para i<-1 até 50 passo 1 faça  
 contador[i] <- 0  
 fim para

para i<-1 até 50 passo 1 faça  
 escreva "Digite a altura: "  
 leia alturas[i]

para x<-1 até 50 passo 1 faça

se (alturas[i] == alturas[x]) então  
 contador[i] <- contador[i] + 1

se(contador[i]>maisRepitido) então  
 maisRepitido<- alturas[i]  
 fim se  
 fim se  
 fim para

soma<- soma + alturas[i]  
 fim para

media<- soma/50  
 desvioPadrao <-((soma \* soma)/50) - (media \* media)  
 mediana <- (alturas[25] + alturas[26])/2

escreva "Media: ", media, " | Desvio padrão: ", desvioPadrao, " | Moda: ",   
 maisRepitido, " | Mediana: ", mediana

fim