Variável Composta

Diferente das variáveis simples, que não seguem uma ordem, apenas reservam espaços na memória do computador conforme a necessidade do sistema operacional.

Variável simples exemplo:

Var

N1: Inteiro

N2, N3, N4: Inteiro

Variável composta exemplo:

Var

N: vetor (1..4) de inteiro

No exemplo da variável composta, ao invés de criar 4 variáveis para reservarem espaço na memória como as variáveis simples, a variável composta reserva 4 espaços na memória, os espaços estão em ordem de 1 até 4 e em sequência, diferente das variáveis simples que reservam espaços aleatórios na memória.

As variáveis compostas tem esse nome pois são compostas de vários espaços para armazenar dados.

**O que é um vetor em variáveis?**

Vetor é a quantidade de espaço que a variável composta vai reservar.

**Atribuição de valores em variáveis compostas**

Ex1:

Algoritmo “atribuição”

Var

N: vetor (1..4) de inteiro

Inicio

N [1] <- 3

N [2] <- 5

N [3] <- 1

N [4] <- 0

FimAlgoritmo

Variáveis Compostas Homogêneas X Variáveis Compostas Heterogêneas

As variáveis compostas homogêneas recebem esse nome pois todos os espaços reservados pela variável recebem o mesmo tipo de dado (Inteiro).

Variáveis Compostas Unidimensionais

São chamadas assim pois ocupam somente 1 endereço (que seria o da variável N).

Ex2:

Algoritmo “atribuição1.0”

Var

N: vetor (1..4) de inteiro

C: Inteiro

Inicio

Para C <- 1 ate 4 faca

Escreva (“Digite o “, C, “o valor: “)

Leia (N [i])

FimPara

Para C <- 1 ate 4 faca

Escreva (“[“, N [C], “]”)

FimPara

FimAlgoritmo

Nesse exemplo, conforme vai mudando o valor da variável i na estrutura de repetição, o usuário vai inserir o valor em determinado espaço ocupado pesa variável N, começando pelo espaço 1 e encerrando no espaço 4.

Ex3:

Algoritmo “ValoresPares”

Var

N: vetor [1..7] de inteiro

C: Inteiro

NPares: Inteiro

Inicio

NPares <- 0

Para C <- 1 ate 7 face

Escreva (“Digite o “, C, “o valor: “)

Leia (N [C])

FimPara

Para C <- 1 ate 7 faca

Se (N [C] % 2 = 0) Entao

NPares <- NPares + 1

Escreval (“O “, C, “o valor inserido é par”)

FimSe

FimPara

Escreval (“Você escreveu “, NPares, “ números pares”)

FimAlgoritmo

Ex4:

Algoritmo “Turma”

Var

Nome: vetor [1..5] de Caractere

C, AlunosAprov, AlunosReprov, AlunosAcimaMedTurma: Inteiro

N1: vetor [1..5] de Inteiro

N2: vetor [1..5] de Inteiro

N3: vetor [1..5] de Inteiro

N4: vetor [1..5] de Inteiro

Media: vetor [1..5] de Real

SomaMed, MedTurma: Real

Inicio

AlunosAprov <- 0

AlunosReprov <- 0

AlunosAcimaMedTurma <- 0

Para C <- 1 ate 5 faca

Escreva (“Escreva o nome do aluno (a): “)

Leia (Nome [C])

Escreval (“Qual foi a primeira nota dele? “)

Leia (N1 [C])

Escreval (“Qual foi a segunda nota dele? “)

Leia (N2 [C])

Escreval (“Qual foi a terceira nota dele? “)

Leia (N3 [C])

Escreval (“Qual foi a quarta nota dele? “)

Leia (N4 [C])

Media [C] <- N1 [C] + N2 [C] + N3 [C] + N4 [C] /4

Escreval (“A soma das notas do aluno “, Nome, “ foi “, Media”

Se (Media >= 6.0) Entao

AlunosAprov <- AlunosAprov + 1

Escreval (“Aluno Aprovado”)

SeNao

AlunosReprov <- AlunosReprov + 1

Escreval (“Aluno Reprovado”)

FimSe

FimPara

LimpaTela

Escreval (“L I S TA G E M D E A L U N O S”)

Escreval (“------------------------------------------“)

Para C <- 1 ate 5 faca

Escreval (“Nome: “, Nome, [C] :15 “ Média: “, Media [C] : 4 : 1)

FimPara

Escreval (“------------------------------------------“)

Escreval (AlunosAprov, “ aluno (s) foram Aprovados e “, AlunosReprov, “ aluno (s) foram Reprovados)

SomaMed <- Media [1] + Media [1] + Media [1]

MedTurma <- SomaMed / 4

Para C <- 1 ate 5 faca

Se (Med [C] > MedTuma) Entao

AlunosAcimaMedTurma <- AlunosAcimaMedTurma + 1

FimSe

FimPara

Escreval (AlunosAcimaMedTurma, “ alunos estão acima da média da turma que é “, SomaMed : 4 : 1)

FimAlgoritmo

Para organizarmos a apresentação de uma lista caso desejemos que o item a seguir de uma variável na lista fique na meta reta que os itens que ocupam a mesma coluna que ela, devemos colocar após a varíavel (quando formos exibir ela na tela) : quantidade de espaços que devem possui mesmo que o tamanho da variável seja menor (assim os espaços restantes serão completados por espaços em branco).

Ex5:

Var

Nome <- Caractere

Inicio

Nome <- David

Escreval (Nome : 15, “ oi”)

FimAlgoritmo

Ex6:

Algoritmo “MostrarNomeComComecoC”

Var

Nome: vetor [1..6] de Caractere

C, CNomeComC: Inteiro

Inicio

CNomeComC <- 0

Para C <- 1 ate 6 faca

Escreva (“Digite um nome: “)

Leia (Nome [C])

Se (Copia (Nome [C], 1, 1) = “C”) Entao

CNomeComC <- CNomeComC + 1

FimSe

FimPara

Escreval (CNomeComC, “ nomes de pessoas começam com a letra C”)

Escreval (“Lista de pessoa com a letra C no começo do nome”)

Escreval (“------------------------------------------------------------------“)

Para C <- 1 ate 6 faca

Se (Copia (Nome [C], 1, 1) = C) Entao

Escreval (“O “, C, “o nome digitado: “,Nome [C])

FimSe

FimPara

Escreval (“------------------------------------------------------------------“)

FimAlgoritmo

Ex7:

Algoritmo “MostramEmOrdem”

Var

N: vetor [1..4] de Inteiro

C, C2, Aux: Inteiro

Inicio

Para C <- 1 ate 4 faca

Escreva (“Digite um número: “)

Leia (N [C])

FimPara

Para C <- 1 ate 3 faca

Para C2 <- C + 1 ate 4

Se (N [C] > N [C2] Entao

Aux <- N [C]

N [C] <- N [C2]

N [C2] <- Aux

FimSe

FimPara

FimPara

Para C <- 1 ate 4 faca

Escreva (“(“, N [C], “)”

FimAlgoritmo

Nesse exemplo após inserirmos os números entramos em uma estrutura de repetição onde na primeira irá inciar com 1 e na segunda com sempre +1 que a primeira. Sempre que o segundo número avaliado na comparação (N [C2]) for menor que o primeiro (N [C]), ocorre a troca, mas para isso, é necessário passarmos o valor da primeira variável para outra varável (Aux) para que seu espaço reservado não esteja ocupado e a informação que estava presente ali não seja perdida, após isso passamos o valor do segundo número (N [C2]) para o primeiro (N [C]) e passando o valor do primeiro número que foi armazenado em outra variável (Aux) para o segundo número (N [C2]). Com isso, caso N [2] seja menor que N [1] os valores de ambos serão trocados, assim, quando N [1] for comparado com N [3], o valor de N [1] será o valor que o N [2] possuía anteriormente, caso N [1] seja maior que N [3] os valores serão trocados, assegurando assim que N [1] fique com o menor valor no final do processo.

Ex8:

Algoritmo “TorneioDeFutebol”

Var

Time <- vetor [1..3] de Caractere

Aux <- Caractere

C, C2, C3 <- Inteiro

Inicio

Escreval (“-------------------------------------“)

Escreval (“CAMPEONATO DE FUTEBOL”)

Escreval (“-------------------------------------“)

Para C <- 1 ate 3 faca

Escreva (“Nome do “, C, “o time: ”)

Leia (Time [C])

FimPara

LimpaTela

Escreval (“-------------------------------------“)

Escreval (“ TABELA DE PARTIDAS ”)

Escreval (“-------------------------------------“)

C <- 1

Enquanto (C <= 3) faca

Para C2 <- 1 ate 2

Para C3 <- C2 + 1 ate 3

Escreva (Time [C2] : 15, “[ ] X [ ]”, Time [C3])

FimPara

Aux <- Time [C3]

Time [C3] <- Time [C2]

Time [C2] <- Aux

FimPara

C <- C + 1

FimEnquanto

FimAlgoritmo

Ex9:

Algoritmo “CorrigindoProva”

Var

Questao: vetor [1..5] de Caractere

Aluno, Resp1, Resp2, Resp3, Resp4, Resp5: vetor [1..5] de Caractere

Nota: vetor [1..5] de Inteiro

C, NotaBoa: Inteiro

Inicio

NotaBoa <- 0

Escreval (“Escreva o Gabarito da Prova”)

Para C <- 1 ate 5 faca

Escreva (“Questão “, C, “:” )

Leia (Questao [C])

FimPara

Para C <- 1 ate 5 faca

Nota [C] <- 0

Escreva (“------------------------------------“)

Escreval (“Aluno “, C)

Escreval (“------------------------------------“)

Escreval (“Nome: “)

Leia (Nome [C])

Escreval (“Respostas Dadas”)

Escreval (“Questão 1:” )

Leia (Resp1 [C])

Se (Resp1 [C] = Questao [1])

Nota [C]<- Nota [C] + 1

FimSe

Escreval (“Questão 2:” )

Leia (Resp2 [C])

Se (Resp2 [C] = Questao [2])

Nota [C]<- Nota [C] + 1

FimSe

Escreval (“Questão 3:” )

Leia (Resp3 [C])

Se (Resp3 [C] = Questao [3])

Nota [C]<- Nota [C] + 1

FimSe

Escreval (“Questão 4:” )

Leia (Resp4 [C])

Se (Resp4 [C] = Questao [4])

Nota [C]<- Nota [C] + 1

FimSe

Escreval (“Questão 5:” )

Leia (Resp5 [C])

Se (Resp5 [C] = Questao [5])

Nota [C]<- Nota [C] + 1

FimSe

Nota [C] <- Nota [C] \* 2

FimPara

Escreval (“Nota dos alunos: ”)

Para C <- 1 ate 5 faca

Escreval (“O aluno “, Nome [C] : 15, “ tirou “, Nota [C], “ na prova”)

Se (Nota [C] < 6)

Escreval (”O aluno “,Nome [C], “ tirou “, Nota [C], “ então terá que fazer recuperação”)

SeNao

screval (”O aluno “,Nome [C], “ tirou “, Nota [C], “ então não terá que fazer recuperação”)

NotaBoa <- NotaBoa + 1

FimSe

FimPara

Escreval (NotaBoa, “ foram acima da média na prova, parabéns”)

FimAlgoritmo

Ex10:

Algoritmo “ResevarDeAssentos”

Var

B: vetor [1..10] de Inteiro

C: vetor [1..10] de Inteiro

R: Caractere

Lugar: Caractere

Inicio

Repita

Escreval (“-----------------------------------------------------“)

Para C <- 1 ate 10 faca

Se (B [C] = 1) Entao

Escreva (“[ ---- ]”)

SeNao

Escreva(“[ “, B [C], “ ]”)

FimSe

FimPara

Escreval (“-----------------------------------------------------“)

Escreval (“Reservar a cadeira: B”)

Leia(Lugar)

Se (B [Lugar] <> 1) Entao

B [Lugar] <- 1

Escreval (“Cadeira B”, Lugar, “ Reservada”)

SeNao

Escreval (“A cadeira B”, Lugar, “ já está Reservada”)

FimSe

Escreval (“Quer reservar outra lugar? [S/N]” )

Leia (R)

Ate (R = S)

FimAlgoritmo