

Gestão de Tecnologia da Informação

Algoritmos

Prof Carlos Eduardo Câmara (dinho)

Aula 05



1

Gestão de Tecnologia da Informação

Algoritmos

Prof Carlos Eduardo Câmara (dinho)

Aula 08



2

GIT - Algoritmos***exercícios:***

Determine os resultados obtidos: $a = 2$; $b = 7$ e $c = 3,5$, L é falso

a) $b > a$ **ou** $b = \text{pot}(a, a)$

b) L **ou** $\text{pot}(b, a) \leq c * 10 + a * b$

c) $\text{pot}(2, 4) \leq 4 + 2$ **ou** $2 + 3 * 5 / 3 \bmod 5 < 0$

d) leia os dados de entrada: x , y , casa, endereço
nome, rua, número, apto,

e) imprima os dados: “meu nome é:”, nome, “meu sobrenome é:” snome

3

GIT - Algoritmos***estruturas de controle – sequencial:***

estrutura sequencial

em um algoritmo corresponde a executar o conjunto de ações segundo uma sequência linear de cima para baixo e, da esquerda para adireita

ações são seguidas por ponto e vírgula, conforme visto anteriormente

isso ajuda a separar estas ações uma das outras e a organizar a sequência de execução:

modelo geral:

1 – inicio. // inicio do bloco ou algoritmo	8 – ação 3
2 –	9 – ...
3 – // declaração de variaveis	10 – ação n
4 –	11 – ...
5 – corpo do algoritmo	12 – fim // fim do bloco ou do algoritmo
6 – ação 1	
7 – ação 2	

4

GIT - Algoritmos***estruturas de controle :***

exemplo:

construa um algoritmo que calcule a média aritmética entre quatro notas quaisquer, fornecidas por um aluno (usuario).

dados de entrada: quatro notas –

dados de saída: média aritmética das notas

qual o modelo para obter a média das quatro notas?

média aritmética

como se calcula a média aritmética?

(...) / 4

lembrar que nota é um número!

5

GIT - Algoritmos***estruturas de controle :***

1. inicio // começo do algoritmo
2. // declaração das variáveis
3. real: n1, n2, n3, n4, mf; // notas bimestrais e média final
4. // entrada de dados
5. leia(n1, n2, n3, n4);
6. $mf \leftarrow (n1 + n2 + n3 + n4)/4$;
7. // saída de dados
8. escreva(mf);
9. // melhorando
10. escreva("nota final: ", mf);
11. fim // término do algoritmo

6

GIT - Algoritmos***estruturas de controle:***

construa um algoritmo que calcule a quantidade necessária de latas de tinta e o custo o custo para pintar tanques cilíndricos de combustível, em que são fornecidos a altura e o raio deste cilindro:

sabendo que:

lata de tinta custa R\$ 50,00

cada lata contém 5 litros

cada litro pinta 3 metros quadrados

dados:

entrada: altura e raio

saída: custo (c), quantidade (qtde)

como resolver? Pode-se partir dos dados de saída! – pensar em como obtê-lo a partir dos dados de entrada!

7

GIT - Algoritmos***estruturas de controle:***

planejamento reverso

custo: quantidade de latas de tinta * custo

quantidade de latas de tintas: total de litros / 5

quantidade de latas de tinta: área do cilindro / 3

área do cilindro: área da base + área lateral

área da base: $\pi * \text{pot}(R, 2)$

área lateral: altura * comprimento: $(2 * \pi * R * H)$

dados raio e altura

π é constante: $\pi = 3.1415$

construir o algoritmo

8

GIT - Algoritmos***estruturas de controle:***

- 1 – inicio
- 2 – declara variáveis
- 3 – entrada de dados
- 4 – calculo da área
- 5 – calculo dos litros
- 6 – calculo da quantidade de latas
- 7 – custo
- 8 – saída de dados
- 9 – fim

9

GIT - Algoritmos***estruturas de controle:***

*

10

Gestão de Tecnologia da Informação

Algoritmos

Prof Carlos Eduardo Câmara (dinho)
Aula 09



11

GIT - Algoritmos

estruturas de controle:

exercício para recordar:

construa um algoritmo que calcule a distância entre dois pontos, dado que a entrada é $P(x_1, y_1)$ e $Q(x_2, y_2)$ e imprima a distância entre eles!

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \leftarrow \text{distância entre dois pontos no plano}$$

12

GIT - Algoritmos***estruturas de controle – seleção :***

permite escolha de um grupo de ações (bloco) a ser executado quando determinada condição ou condições, representadas por expressões lógicas ou relacionais são satisfeitas, ou não!

seleção simples:

se <condição>

então

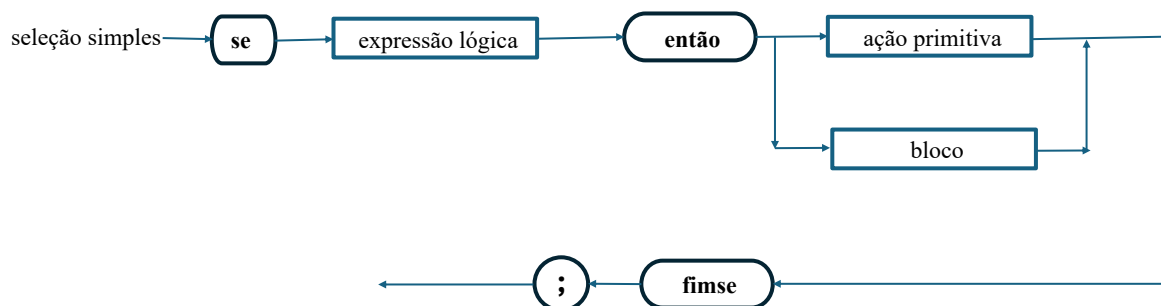
<comando>; // comando único

fimse;

13

GIT - Algoritmos***estruturas de seleção:***

diagrama:



14

GIT - Algoritmos***estruturas de controle – seleção:***

com vários comandos:

se <condição>

então

inicio // inicio do bloco verdade

<comando1>; // comando único

<comando2>; // comando único

...

<comandon> // comando único

fim // fim do bloco verdade

fimse;

15

GIT - Algoritmos***estruturas de controle – seleção:***

exemplo: no algoritmo que calcula média, adicionar: se a média for maior ou igual a 7, aluno está aprovado!

16

GIT - Algoritmos***exemplo:*** média aritmética com aprovação

```

1. inicio
2.  real: n1, n2, n3, n4, mf;      // notas bimestrais, nota final
3.  leia(N1, n2, n3, n4);        // entrada de dados
4.  mf ← (n1 + n2 + n3 + n4) / 4; // processamento
5.  escreva("média final: ", mf); // saída de dados
6.  se (mf >= 7)
7.      então
8.          escreva("Aluno Aprovado");
9.  fimse;
10. fim

```

17

GIT - Algoritmos***estruturas de seleção:***

seleção composta:

permite uma situação onde duas alternativas dependem de uma mesma condição, ser verdadeira e ser falsa ou vice-versa

```

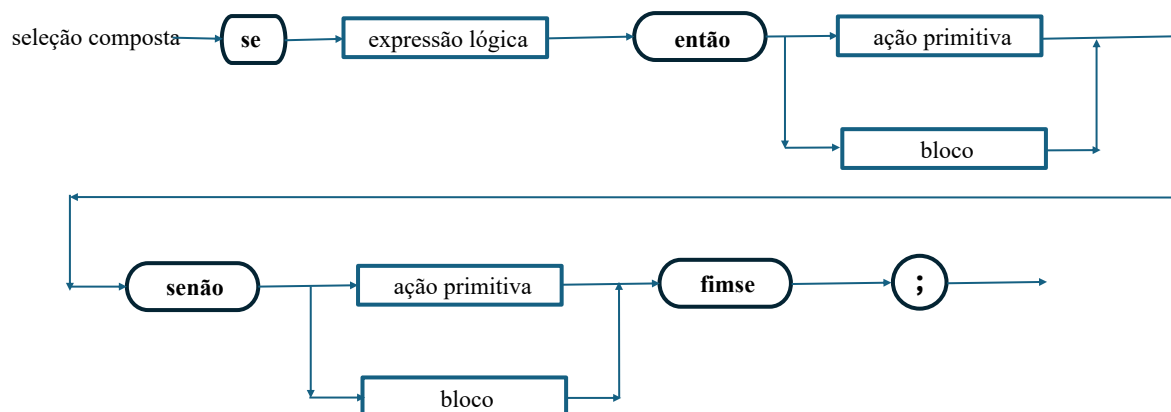
se <condição>
    então
        início
            <comando1> // sequência comando
            <comando2> // sequência comando
            ...
            <comandon> // sequência comando
        fim
    senão
        <comando> // sequência comando
fimse

```

18

GIT - Algoritmos***estruturas de seleção:***

diagrama:



19

GIT - Algoritmos***estruturas de seleção:***

exercício:

construa um algoritmo que calcule as raízes de uma equação do segundo grau do tipo:

 $ax^2 + bx + c = 0$, onde os valores a , b e c são dados fornecidos pelo usuário e as raízes são apresentadas como resposta!

20

GIT - Algoritmos***estruturas de controle – seleção :***

exercício:

início

real: a, b, c, delta, // coeficientes da equação e delta

x1, x2; // raízes da equação

leia(a, b, c) // coeficientes da equação

delta = $\text{pot}(b, 2) - 4 * a * c$;

se (d > 0)

então

início

x1 $\leftarrow (-b + \text{rad}(d)) / 2 * a$;x2 $\leftarrow (-b - \text{rad}(d)) / 2 * a$;

escreva (“raíz x1 = “, x1, “ e raíz x2 = “,

x2);

fim;

senão

se (d = 0)

então

início

escreva(“ Delta = 0, a equação possui uma única raíz! “, /n);

x1 = $(-b) / 2 * a$;

escreva(“raíz x1 = “, x1);

fim;

senão

escreva(“ não possui raízes reais!”);

fimse;

fimse;

fim

21

GIT - Algoritmos***estruturas de seleção:******seleção composta:*** no caso de existir um conjunto de ações que deveria ser executado quando o resultado da condição for falso, cria-se onbloco falsidade

se <condição>

então

início

<comando1>; // sequência comando

<comando2>; // sequência comando ...

<comandon>; // sequência comando

fim;

senão

início

<comando1>; // sequência comando. ...

<comandon>; // sequência comando n

fim;

fimse;

22

GIT - Algoritmos***exemplo:***

escrever o algoritmo de média aritmética com aprovação e reprovação

23

GIT - Algoritmos***exemplo:*** média aritmética com aprovação

```

1. inicio
2.   real: n1, n2, n3, n4, mf;           // notas bimestrais, nota final
3.   leia(N1, n2, n3, n4);              // entrada de dados
4.    $mf \leftarrow (n1 + n2 + n3 + n4) / 4$ ; // processamento
5.   escreva("media final: ", mf);       // saída de dados
6.   se (mf >= 7)
7.     então
8.       escreva("Aluno Aprovado");
9.     senão
10.      escreva("Aluno Reprovado ");
10. fimse;
10. fim
```

24

GIT - Algoritmos***estrutura de controle – seleção encadeada:***

<i>se</i> <condicao1>	inicio	
<i>então</i>	c1	
<i>se</i> <condicao2>	
<i>então</i>	cn	
inicio	fim	
c1;	<i>senão</i>	
.....	cf	// comando falsidade
cn;	<i>fimse</i> ;	
fim	<i>fimse</i> ;	
<i>fimse</i>	
<i>senão</i>		
<i>se</i> <condicao3>		
<i>então</i>		

25

GIT - Algoritmos***exemplo:***

escrever o algoritmo de média aritmética com aprovação ($\geq 7,0$), reprovação ($\leq 5,0$) e recuperação quando a nota estiver entre 5,0 e 7,0

26

Gestão de Tecnologia da Informação

Algoritmos

Prof Carlos Eduardo Câmara (dinho)

Aula 10

exercícios



27

GIT - Algoritmos

exercício 1:

dados três valores A, B, C, verifique se eles podem ser os comprimentos dos lados de um triângulo. Se forem, verifique se compõem um triângulo equilátero, isósceles, ou escaleno. Informe se não compuserem triângulo nenhum.

dados de entrada A, B, C – três supostos lados de um triângulo (A, B, C)

dados de saída: mensagens: não compõem triângulo, triângulo equilátero, triângulo isósceles e triângulo escaleno

triângulo – figura geométrica fechada de três lados, em que cada um é menor que a soma dos outros dois

triângulo equilátero – triângulo com os três lados iguais

triângulo isósceles – um triângulo com dois lados iguais

triângulo escaleno – triângulo com todos os lados diferentes

28

GIT - Algoritmos***exemplo:***

é triângulo se: $(A < B + C)$ e $(B < A + C)$ e $(C < A + B)$

é equilátero se: $(A = B)$ e $(B = C)$

é isóceles se: $(A = B)$ ou $(B = C)$ ou $(A = C)$

é escaleno se: $(A > B)$ e $(B > C)$ e $(A > C)$

construir o algoritmo

29

GIT - Algoritmos***exemplo - continua:***

é triângulo	é equilátero	é isóceles	é escaleno	ações
V	V	F	F	“equilátero”
V	F	V	-	“isóceles”
V	F	F	V	“escaleno”
F	-	-	-	“não é triângulo”

30

GIT - Algoritmos

e:

31

GIT - Algoritmos

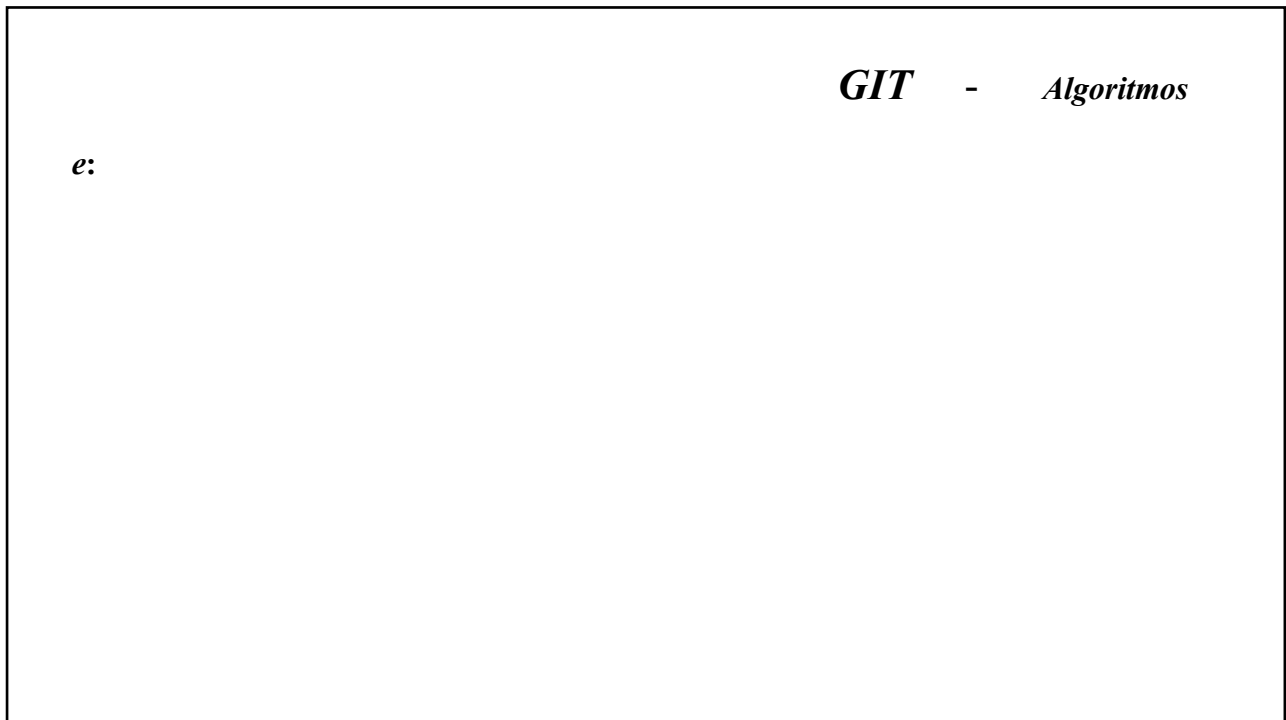
exercício 2:

Entre com dia e mês e devolva o signo do zodiaco para a referida data!
pesquise as datas de cada signo!

exercício 3:

Dada a idade de uma pessoa, entrada pelo usuário, elaborar um algoritmo que a classifique como:
sendo menor que 16 anos não vota, (não votante), entre 18 e 65 voto obrigatório e maior de 65, voto facultativo.

32



33



34

GIT - Algoritmos**estruturas de repetição:**

repetição com teste no início

é uma estrutura de controle de fluxo de execução que permite repetir diversas vezes um mesmo trecho de algoritmo, porém sempre verificando de cada execução se é permitido executar o mesmo trecho

enquanto <condição> **faça**

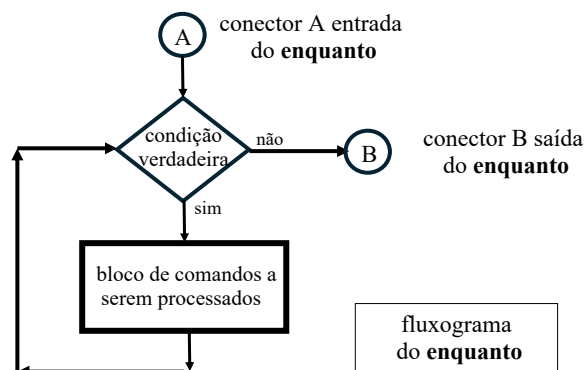
c1;

c2;

.....

cn;

fimenquanto;



35

GIT - Algoritmos**exercício:**

inserir o cálculo da média dos alunos em um laço de repetição, usando a estrutura do enquanto (while)

inicio

real: n1, n2, n3, n4, mf;

inteiro: con; // contador ou inteiro: con ← 0

con = 0;

enquanto (con < 50) **faça** // teste de condição de parada

leia(n1, n2, n3, n4);

mf ← (n1 + n2 + n3 + n4) / 4;

escreva("média final = ", mf);

.... // programa continua com as comparações

con = con + 1;

fimenquanto;

fim

36

exercício:

construa um algoritmo que calcule
 $ax^2 + bx + c = 0$, onde os valores a ,
 apresentadas como respos
 utilize a repetição enquanto para re

```
FatecCampinas_2024 > aula04 > baskara_algor.por > baskara_alg_aula.por
1  inicio
2      real: a, b, c, delta, x1, x2;
3
4      escreva("entre com os coeficientes: a, b, c: ");
5      escreva("entre com o coeficiente a = ");
6      leia(a);
7      escreva("entre com o coeficiente b = ");
8      leia(b);
9      escreva("entre com o coeficiente c = ");
10     leia(c);
11
12     delta = (pot(b, 2)) - 4 * a * c
13     escreva("delta = ", delta)
14     se (delta > 0)
15     {
16         x1 = (-b + rad(delta)) / 2 * a
17         x2 = (-b - rad(delta)) / 2 * a
18         escreva("a equação possui duas raízes, x1 = ", x1, "e x2 = ", x2)
19     }
20     senao
21     {
22         se ( delta == 0)
23         {
24             x1 = (-b) / 2 * a
25             escreva("a equação possui uma única raiz real x = ", x1)
26         }
27         senao
28         {
29             escreva(" equação não possui raízes reais")
30         }
31     }
32 }
33
34 fim
```

37

GIT - Algoritmos

exercício:

construa um algoritmo que calcule as raízes de uma equação do segundo grau do tipo:
 $ax^2 + bx + c = 0$, onde os valores a , b e c são dados fornecidos pelo usuário e as raízes são
 apresentadas como resposta!
 utilize a repetição enquanto para resolver várias equações!

38

GIT - Algoritmos***exercício:***

construir um algoritmo que calcule a média aritmética dos 50 alunos de algoritmos, continuação do exercício anterior...., notas: real: n1, n2, n3, n4, mf, media_de_todos_alunos

39

exercício:

construir um algoritmo que calcule a média aritmética dos 50 alunos de algoritmos, continuação do exercício anterior..

```
FatecCampinas_2024 > aula04 > baskara_algor.por > media50alunos.por
1  inicio
2      real: n1, n2, n3, n4, ma, media_todos, media50al;
3      inteiro: con;
4      con = 0;
5      media_todos = 0;
6      media50al = 0;
7      escreva("este algoritmo calcula a média de 50 alunos")
8      enquanto(con < 50)
9          faça
10             escreva(" entre com a nota do aluno número: ", con);
11             escreva("entre com a nota n1 = ");
12             leia(n1);
13             escreva("entre com a nota n2 = ");
14             leia(n2);
15             escreva("entre com a nota n3 = ");
16             leia(n3);
17             escreva("entre com a nota n4 = ");
18             leia(n4);
19             mf = (n1 + n2 + n3 + n4)/4;
20             escreva("média do aluno número: ", con, " é: ", mf);
21             media_todos = media_todos + mf;
22             con = con + 1;
23         fimenquanto;
24         media50al = media_todos/50;
25         escreva("a media dos 50 alunos é: ", media50al);
26     fim
```

40

GIT - Algoritmos***repetição com teste final:***

construir uma estrutura **repita** dentro da qual a ação primitiva ou bloco repita **até** que determinada condição seja verdadeira

repita:

início

<condição 1>;

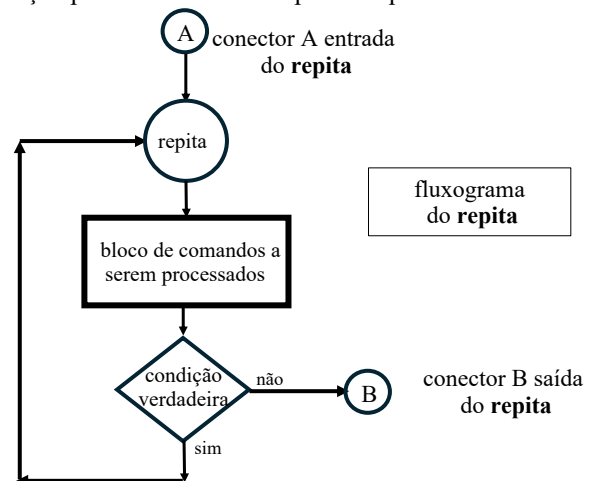
<condição 2>;

<condição 3>;

...

<condição n>;

até <condição>



41

GIT - Algoritmos***exercícios:***

refaça o exercício calcular a média aritmética final de 50 alunos, usando *repita*, *até*...
siga o algoritmo passo a passo e verique a diferença entre os dois tipos de estruturas!

42

GIT - Algoritmos***repetição com variável de controle:***

é uma estrutura de repetição onde é possível determinar o número de vezes que um laço será percorrido (repetido)

para (i de vi até vf, passo p)

faça

início

<condição 1>;

<condição 2>;

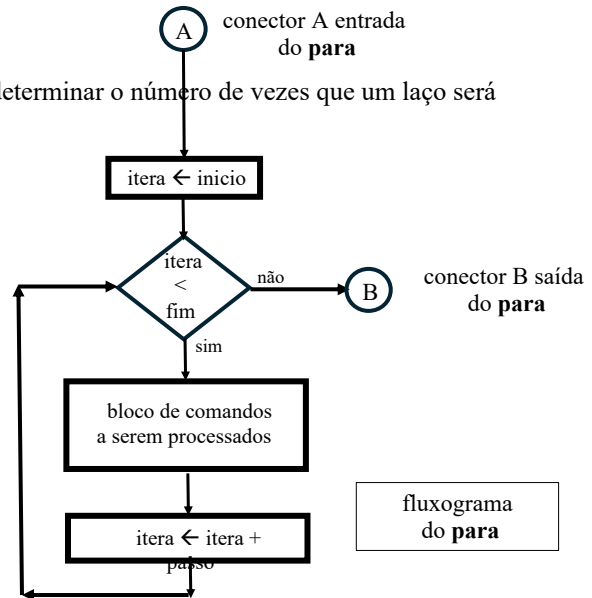
<condição 3>;

...

<condição n>;

fim;

fimpara;



43

GIT - Algoritmos***exercício:***

construir um algoritmo utilizando esta estrutura **para ... fimpara** para calcular a média total de 50 alunos

44

GIT - Algoritmos***Comparação entre as estruturas de repetição:***

estrutura	condição de parada	quantidade de execuções	condição de existência
<i>enquanto</i>	início	0 ou muitas	verdadeira
<i>repita</i>	final	Mínimo uma	falsa
<i>para</i>	não tem	$((vf - vi) \div p) + 1$	$v <= vf$

45

GIT - Algoritmos***exercício:***

A sequência de Fibonacci tem uma formação do tipo: 1, 1, 2, 3, 5, 8, 13, 21, 34, ...

elaborar um modelo de crescimento para esta sequência e construir um algoritmo que implemente este modelo.

Pense e desenvolva usando as estruturas *repita ... até*, e *para - faça* -

46

GIT - Algoritmos***exercício:***

construa um algoritmo que seja capaz de concluir qual dentre os seguintes animais foi escolhido, através de perguntas e resposta. Animais possíveis:

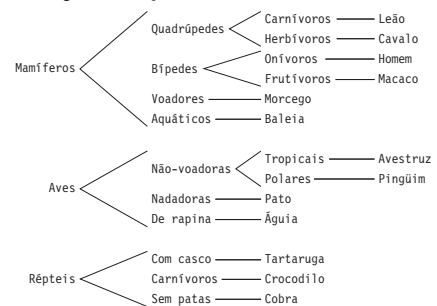
leão, macaco, morcego, baleia, avestruz, pinguim, crocodilo, águia, e cobra

Exemplo

É mamífero? Sim.
É quadrúpede? Sim.
É carnívoro? Não.
É herbívoro? Sim.

Então o animal escolhido foi o cavalo.

Utilize as seguintes classificações:



47

GIT - Algoritmos***exercício:***

1- em uma eleição presidencial existem quatro candidatos. Os votos são informados em código. Os dados informados para a escrutinagem obedecem a seguinte codificação:

. 1, 2, 3, 4 – voto para os respectivos candidatos

. 5 – voto nulo

. 6 – voto em branco

desenvolva um algoritmo que calcule escreva:

. total de votos para cada candidato e seu percentual sobre o total

. total de votos nulos e seu percentual sobre o total

. total de votos em branco e seu percentual sobre o total

para finalizar o conjunto de votos, o valor de entrada deve ser 0 (zero)

48

GIT - Algoritmos***exercício:***

escreva um algoritmo que imprima todos os números primos existentes entre dois números de entrada, n1 e n2, números naturais fornecidos pelo usuário

49

GIT - Algoritmos***exercício:***

2- dado o algoritmo, responda:

- a) o que será mostrado se os números forem 4 e 0
- b) o que será mostrado se os números forem 3, 2 e 2
- c) o que será mostrado se os números forem 2, 1, e 0

```
FatecCampinas_2024 > aula04 > baskara_algor.por > exerc3forb.por
1  inicio
2      inteiro: a, b, i, j;
3      leia(a);
4      repita
5          para i de 1 até a, passo 1, faça
6              j <-- i;
7              enquanto (j <= a) faça
8                  escreva(j);
9                  j <-- j + 1;
10             fimenquanto;
11         fimpara;
12         b <-- a;
13         leia(a)
14     até ((a = b) ou (a <= 0));
15 fim.
```

50

GIT - Algoritmos***exercício:***

3- elabore um algoritmo que imprima ou escreva todos os números primos entre n1 e n2 que são números definidos pelo usuário.

4- calcule o imposto de renda de um grupo de dez contribuintes. É fornecido o valor do salário mínimo no início dos cálculos. Cada contribuinte entra com o CPF, renda mensal, número de dependentes. Para cada dependente o contribuinte tem um desconto de 5% do salário mínimo.

os valores da alíquota para cálculo do IP são:

renda líquida	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%

51

GIT - Algoritmos***exercício:***

52

GIT - Algoritmos***exercício:***

5- construa um algoritmo que com a entrada da altura, peso e sexo de uma pessoa, calcule seu peso ideal, utilizando as fórmulas:

para homens: $(72.7 * h) - 58.0$

para mulheres: $(62.1 * h) - 44.7$

escreva saída do cálculo para 10 pessoas e calcule a média dos pesos dos homens e das mulheres separadamente que entraram com seus dados.

53

GIT - Algoritmos***exercício:***

6- elabore um algoritmo que leia dois números inteiros e a operação aritmética desejada. Calcule então a resposta adequada. Operações aritméticas: +, -, *, /.

7- Calcule o IMC da massa corporal para dar uma indicação sobre a condição de peso de uma pessoa adulta. A fórmula do IMC $\text{peso}/(\text{altura})^2$.

IMC em adultos	condição
Abaixo de 18,5	Abaixo do peso
Entre 18,5 e 25	Peso normal
Entre 25 e 30	Acima do peso
Acima de 30	obeso

54

GIT - Algoritmos***exercício:***

- 8) calcule a tabuada de qualquer número de 0 a 10;
- 9) em um prédio há três elevadores, A, B e C. Para otimizar o sistema de controle dos elevadores foi realizado um levantamento no qual cada usuário respondia:
- . elevador que utiliza com mais frequência
 - . período que utiliza o elevador com mais frequência: M = matutino, V = vespertino e N = noturno
- construa um algoritmo que calcule e imprima:
- qual elevador é mais utilizado e em que período se concentra essa utilização
- qual o período mais usado de todos e a qual elevador pertence
- qual a diferença percentual entre o mais usado e o menos usado no horário M

55

GIT - Algoritmos***bibliografia:***

FORBELLONE, ELBERSPÄTCHER, Logica de Programação, 3.a e 4.a Ed. Pearson – Bookman, 2022

https://www.ime.usp.br/~leo/mac2166/2017-1/line_introducao_while.html, prof. L. O. Brandão –

LINE.USP.BR – em 02/07/2024 (fluxogramas para e enquanto)

MENEZES, Introdução à Programação com Python, 4.a ed, Novatec, 2024

56

bibliografia:FORBELLONE, EBERSPÄTCHER, *Lógica de Programação*, 3

```

1 programa
2 {
3     inclui biblioteca Matematica --> mat
4     funcao inicio()
5     {
6         real a, b, c, delta, x1, x2
7         escreva("entre com os coeficientes da equação: ax² + bx + c = 0: ")
8         escreva("entre com a: ")
9         leia(a)
10        escreva("entre com b: ")
11        leia(b)
12        escreva("entre com c: ")
13        leia(c)
14
15        delta = (mat.potencia(b, 2.0) - 4 * a * c)
16
17        se (delta > 0 )
18        {
19            escreva("a equação tem duas raízes reais")
20            x1 = (-b + mat.raiz(delta, 2.0))/ 2*a
21            x2 = (-b - mat.raiz(delta, 2.0))/ 2*a
22            escreva("as raízes são: x1 = ", x1, " e x2 = ", x2)
23        }
24        senao
25        { se ( delta == 0)
26        {
27            escreva("a equação tem apenas uma raiz real!")
28            x1 = (-b)/ 2*a
29            escreva("a raiz única é x = ", x1)
30        }
31        senao
32        {
33            escreva("a equação não possui raízes reais!")
34        }
35        }
36    }
37 }
38
39
40

```

PROBLEMS OUTPUT TERMINAL ENTRADA E SAÍDA DEBUG CONSOLE PORTS

entre com os coeficientes da equação: ax² + bx + c = 0: entre com a: > 2
entre com b: > 1
entre com c: > 1
a equação não possui raízes reais!
Fim da execução.

57

Gestão de Tecnologia da Informação

Algoritmos

Prof Carlos Eduardo Câmara
(dinho)

Aula 14 e 15

58

GIT - Algoritmos***estruturas de dados:***

divisão imaginária para simplificar a classificação dos dados
 tipos de dados primitivos não são suficientes para representar todo tipo de dados e de informação
 novos tipos foram construídos, a partir da composição de tipos primitivos
 denominando estrutura de dados que definem a forma de organizar esses tipos primitivos

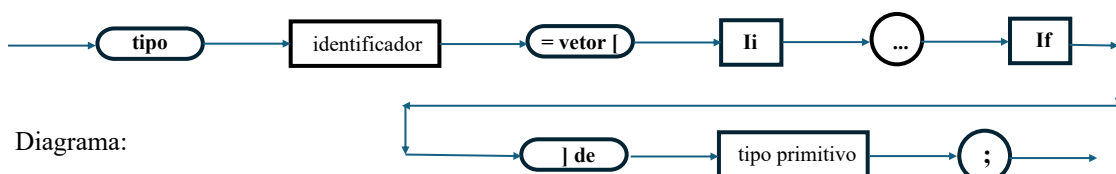
as gavetas agora podem comportar um conjunto de dados, desde que previamente organizadas e compartimentos
 esses novos tipos deve ser definidos em cada detalhe de sua estrutura

59

GIT - Algoritmos***estrutura de dados:***

variáveis compostas homogêneas:
 a variável pode ser interpretada como um elemento de um conjunto, que é a estrutura de dados
 quando a estrutura de dados é composta por elementos de um mesmo tipo primitivo, a estrutura de dados é dita homogênea
 como numa alcatéia onde todos seus elementos são lobos

estrutura composta unidimensional – é uma estrutura denominada vetor unidimensional



60

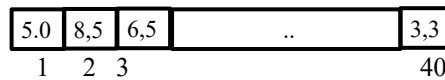
GIT - Algoritmos***estrutura de dados:***

exemplo

um vetor de 50 posições -

tipo classe = **vetor** [1 .. 40] **de reais**; // def do tipo vetor CLASSE: VCLASSE;

// declaração da variável vetor



posição → mas o índice, na maioria das linguagens inicia com (0) zero
logo o índice é 2 e esta posição é representada por $v[2] = 6,5$

```

v[2] ← 6,5;      // atribuir 6,5 a posição 2 do vetor
leia(v[2]);      // leia o valor a ser atribuído a posição 2 do vetor
escreva(v[2]);   // escreva o valor contido na posição 2 do vetor, nesse caso, 6,5

```

61

GIT - Algoritmos***exemplo:***

inicio

inteiro: a, b, c, d, e, f, notasAcima;

real: media;

notasAcima ← 0;

leia(a, b, c, d, e, f);

media ← (a+b+c+d+e+f)/5;

se (a > media)

entao notasAcima ← notasAcima + 1;

fimse;

se (b > media)

entao notasAcima ← notasAcima + 1;

fimse;

.....

fim

se (c > media)

entao notasAcima ← notasAcima + 1;

fimse;

se (d > media)

entao notasAcima ← notasAcima + 1;

fimse;

se (e > media)

entao notasAcima ← notasAcima + 1;

fimse;

se (d > media)

entao notasAcima ← notasAcima + 1;

fimse;

escreva(notasAcima);

62

GIT - Algoritmos

exemplo: se utilizar vetor:

```

inicio
  tipo CLASSE = vetor[1 .. 5] de reais;
  Class: Vclasse; // declaração da variavel composta do tipo definido
  real: soma, media, notaAcima;
  inteiro: i;
  soma  $\leftarrow$  0;
  notaAcima  $\leftarrow$  0;
  para i de 1 ate 5 passo 1 faca
    leia(Vclasse[i]);
  fimpara;

```

63

GIT - Algoritmos

exemplo: se utilizar vetor:

```

// laço para acumular em soma os valores de VClasse
para i de 1 ate 5 passo 1 faca
  soma  $\leftarrow$  soma + Vclasse[i];
fimpara;
media  $\leftarrow$  soma / 5;
// laço para verificar valores de Vclasse acima da média
para i de 1 ate 5 passo 1 faca
  se (Vclasse[i] > media)
    entao notaAcima  $\leftarrow$  notaAcima + 1;
  fimse;
fim para;
escreva(notaAcima);
fim.

```

64

GIT - Algoritmos***exemplo:***

```

soma de dois vetores:
inicio
  tipo v = vetor {1..50} de inteiros;
  v: vetA, vetB, vetC;
  inteiro: i;
  para i de 1 ate 50 passo 1 faca
    leia(vetA[i], vetB[i]);
    vetC[i] ← vetA[i] + vetB[i];
    escreva(vetC[i]);
  fimpara;
fim.

```

65

GIT - Algoritmos***exemplo:***

```

preenchendo vetores
inicio
  tipo vms = vetor [1..100] de inteiros;
  vms: A;
  para i de 1 ate 100 faca
    se ((i mod 2) <> 0)
      entao A[i] ← 1;
      senao A[i] ← 0;
    fimse;
  fimpara
fim.

```

66

GIT - Algoritmos***exercícios:***

escreva um algoritmo que, dados dois vetores de 20 posições, efetue as respectivas operações indicadas por outro vetor de 20 posições de caracteres, também fornecido pelo usuário, contendo as quatro operações em qualquer combinação e armazenando os resultados em um quarto vetor.

altere o exemplo de soma de vetores para que realize a seguinte operação: o produto do primeiro vetor pelo inverso do segundo que é armazenado num terceiro vetor de reais, todos com 20 posições;

desenvolva um algoritmo que leia um vetor de 20 posições e o coloque em ordem crescente utilizando como estratégia de ordenação a comparação entre dois adjacentes, permutando-os quando estiverem fora de ordem, até que esteja ordenados;

67

GIT - Algoritmos***exemplo:***

escreva um algoritmo que leia um vetor com 30 números inteiros e gere um segundo vetor cujas posições pares, são o dobro do vetor original e as ímpares o triplo.

escreva um algoritmo que leia uma série de 50 notas e calcule quantas são 10% acima da média de todas as notas e 10% abaixo.

escreva um algoritmo que coloque como seus elementos os primos de 1 até 100. Calcule a média desses primos.

escreva um algoritmo que permita informar dados para 2 vetores inteiros de 20 posições, apresente um terceiro vetor que tenha o conjunto união dos dois vetores. Lembrando que conjunto união é aquele que possui todos elementos dos dois conjuntos sem repetição. Ou seja, cada elemento aparece uma única vez.

68

GIT - Algoritmos

exemplo: 03/07

- 1) escrever um algoritmo que multiplique dois vetores e coloque o resultado num terceiro vetor. Fazer o fluxograma desse algoritmo.
- 2) calcular os numeros primos entre 2 e 50, colocar em um vetor. Usar um tipo de repetidor que permita alocar apenas a quantidade suficiente de espaços com números. Com Certeza irá sobrar espaços sem valores, mas isso, por enquanto não tem jeito. Fazer um fluxograma sobre o algoritmo.

69

GIT - Algoritmos

FORBELLONE, ELBERSPÄTCHER, Logica de Programação, 3.a e 4.a Ed. Pearson – Bookman, 202
https://www.ime.usp.br/~leo/mac2166/2017-1/line_introducao_while.html, prof. L. O. Brandão –
LINE.USP.BR – em 02/07/2024 (fluxogramas para e enquanto)
MENEZES, Introdução à Programação com Python, 4.a ed, Novatec, 2024

70



71

GIT - Algoritmos

estrutura de dados: variáveis compostas multidimensionais

as matrizes são estruturas compostas multidimensionais

é uma estrutura bidimensional que possui dois índices de referência para definir cada um de seus valores internos

representa-se a matriz como $M[i, j]$ onde i representa o número de linhas e j o número de colunas

$$M = \begin{bmatrix} 1, 1 & 1, 2 & 1, 3 & 1, 4 \\ 2, 1 & 2, 2 & 2, 3 & 2, 4 \\ 3, 1 & 3, 2 & 3, 3 & 3, 4 \end{bmatrix} (3, 4)$$

se $i = j$ a matriz é denominada matriz quadrada

se $i \neq j$ a matriz é denominada matriz retangular,

se $i = 1$ a matriz é matriz linha ou vetor linha, onde $j = 1, 2, 3, 4, 5, \dots$

se $j = 1$ a matriz é matriz coluna ou vetor coluna, onde $i = 1, 2, 3, 4, \dots$

72

GIT - Algoritmos

estrutura de dados: variáveis compostas multidimensionais – matrizes

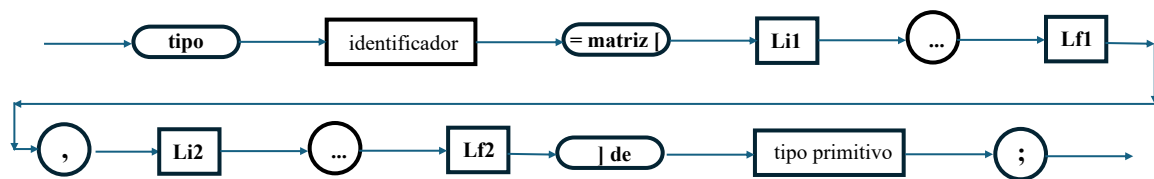
generalizando uma matriz $A = [a_{i,j}]$

é uma matriz com elementos denominados $[a_{i,j}]$

i linhas e j colunas

$$A = \begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} & a_{1,4} \\ a_{2,1} & a_{i,j} & a_{2,3} & a_{2,4} \\ a_{3,1} & a_{3,2} & a_{3,3} & a_{3,4} \end{bmatrix} \quad (3, 4)$$

diagrama da declaração – tipo M = matriz $[i_1..i_2, j_1..j_2]$ de tipo primitivo



73

GIT - Algoritmos

estrutura de dados: variáveis compostas multidimensionais – matrizes

exemplos:

tipo M = matriz [1..3, 2..4, 3..4] de reais; // matriz tridimensional

M: A, B, C;

tipo ID = matriz [1..6, 1..8] de caracteres; // matriz bidimensional

ID: Nome, END, CEP;

dimensão: quantidade de elementos de uma matriz $(6 - 1 + 1) * (8 - 1 + 1) = 48$ elementos – bi-D

quantidade de elementos $(3 - 1 + 1) * (4 - 2 + 1) * (4 - 3 + 1) = 18$ elementos – tri-D

74

GIT - Algoritmos

estrutura de dados: variáveis compostas multidimensionais – matrizes

dado um cartão de loteria como esse ao lado preenchido, desejamos saber qual o jogo que possui mais marcações, ou seja, jogos triplo, ou duplos!

O cartão de loterias possui $i = 14$ linhas; $j = 3$ colunas

$mLoteria[i,1]$, $mLoteria[i,2]$, $mLoteria[i,3]$ onde esse i varia de 1 até 14, construindo essa matriz com um jogo

Jogo	Coluna 1	Empate	Coluna 2
1	<input type="checkbox"/> cxvdbcd	<input type="checkbox"/>	dhghac <input type="checkbox"/>
2	<input type="checkbox"/> qwer	<input type="checkbox"/>	jehgw <input type="checkbox"/>
3	<input type="checkbox"/> rterf	<input type="checkbox"/>	jklopu <input type="checkbox"/>
4	<input type="checkbox"/> erf	<input type="checkbox"/>	hprutwh <input type="checkbox"/>
5	<input type="checkbox"/> oykjytyu	<input type="checkbox"/>	jktyergerg <input type="checkbox"/>
6	<input type="checkbox"/> tytht	<input type="checkbox"/>	swtoih <input type="checkbox"/>
7	<input type="checkbox"/> ijkkyuk	<input type="checkbox"/>	nbmvb <input type="checkbox"/>
8	<input type="checkbox"/> juju	<input type="checkbox"/>	fdfggdgnj <input type="checkbox"/>
9	<input type="checkbox"/> yumyummy	<input type="checkbox"/>	dcwssvv <input type="checkbox"/>
10	<input type="checkbox"/> mfgfffggh	<input type="checkbox"/>	htcwhw <input type="checkbox"/>
11	<input type="checkbox"/> ertrtrtt	<input type="checkbox"/>	rhtrthw <input type="checkbox"/>
12	<input type="checkbox"/> rhghghyk	<input type="checkbox"/>	rhjrh <input type="checkbox"/>
13	<input type="checkbox"/> wrsdd	<input type="checkbox"/>	nhgfhjfgjj <input type="checkbox"/>
14	<input type="checkbox"/> hshfjkihi	<input type="checkbox"/>	fgddhhr <input type="checkbox"/>

75

estrutura de dados: variáveis comp

```

1  inicio
2  // definicao do tipo construido matriz
3  tipo Loteria = matriz [1..14, 1..3] de caracteres;
4  // declaracao da variavel composta do tipo matriz definido
5  Loteria: mLoteria; // nome da matriz
6  inteiro: i, j, maisMar, nJogo, marLin;
7  // indice para linha: I, I; maior marcador encontrado: maisMar
8  // numero de jogos com mais marcações: nJogo
9  // numero de marcacoes em uma linha: marLin
10 maisMar <-- 0;
11 para i de 1 ate 14 faça
12     marLin <-- 0;
13     para j de 1 ate 3 faça
14         se mLoteria[i,j] = "x";
15             então marLin <-- marLin + 1;
16     fimse
17     fimpara
18     se marLin > maisMar
19         então inicio
20             maisMar <-- marLin;
21             nJogo <-- I;
22             fim;
23     fimse;
24 fimpara;
25 escreva("Jogo mais marcado: ", nJogo);
26 escreva("quantidade de marcacoes: ", maisMar);
27 fim.
28

```

76

GIT - Algoritmos

estrutura de dados: variáveis compostas multidimensionais

soma de duas matrizes:

desenvolva um algoritmo que leia duas matrizes A[5, 5]

B[5, 5] e faça a soma!

a soma de duas matrizes é $c[i, j] = a[i, j] + b[i, j]$

```
.vscode > algoritmos > somadematrizes.por
1  inicio
2  tipo M = matriz [ 1..5, 1..5] de inteiros;
3  M; MA, MB, MR;
4  inteiro: i, j;
5  enquanto ( i <= 5) faça
6      j <-- 1;
7      enquanto (j <= 5) faça
8          leia(MA[i,j], MB[i,j]);
9          MR[i,j] <-- MA[i,j] + MB[i,j];
10         j <-- j + 1;
11     fimenquanto;
12     i <-- i + 1;
13 fimenquanto;
14 j <-- j + 1;
15 enquanto ( j <= 5) faça
16     i <-- 1;
17     enquanto (i <=5) faça
18         escreva("MR[" , i, j, "] = ", MR[i,j]);
19         i <-- i + 1;
20     fimenquanto
21     j <-- j + 1;
22 fimenquanto
23 fim.
```

77

GIT - Algoritmos

estrutura de dados: variáveis compostas multidimensionais – matrizes

Representando as matrizes:

$$C = A * B = \begin{bmatrix} c_{11} & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{23} \\ c_{31} & c_{32} & c_{33} \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} * \begin{bmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \end{bmatrix}$$

Onde:

$$c_{11} = a_{11} * b_{11} + a_{12} * b_{21} + a_{13} * b_{31}$$

$$c_{12} = a_{11} * b_{12} + a_{12} * b_{22} + a_{13} * b_{32}$$

$$c_{13} = a_{11} * b_{13} + a_{12} * b_{23} + a_{13} * b_{33}$$

$$c_{21} = a_{21} * b_{11} + a_{22} * b_{21} + a_{23} * b_{31}$$

$$c_{22} = a_{21} * b_{12} + a_{22} * b_{22} + a_{23} * b_{32}$$

$$\text{generalizando } c_{ij} = \sum_{k=1}^3 a_{ik} * b_{kj}$$

78

estrutura de dados: variáveis compostas multidimensionais

algoritmo multiplica $A * B = C$ \rightarrow \rightarrow

```
.vscode > algoritmos > multiplica2mat.por
1  inicio
2      tipo MAT = matriz[1..3, 1..3] de inteiros;
3      MAT: A, B, C; matrizes
4      inteiro: i, j, k;
5      para i de 1 ate 3 passo 1 faça
6          para j de 1 até 3 passp 1 faça
7              leia (A[i,j]);
8              fimpara;
9          fimpara;
10         para i de 1 ate 3 passo 1 faça
11             para j de 1 até 3 passp 1 faça
12                 leia (B[i,j]);
13                 fimpara;
14             fimpara;
15         // calcular C = A*B
16         para i de 1 ate 3 passo 1 faça
17             para j de 1 até 3 passp 1 faça
18                 C[i,j] <-- 0;
19                 para k de 1 ate 3 faça
20                     C[i,j] <-- A[i,k] * B[k,j];
21                 fimpara;
22             fimpara;
23         fimpara;
24         // escrevendo a matriz C resposta
25         para i de 1 ate 3 passo 1 faça
26             para j de 1 até 3 passp 1 faça
27                 escreva(" C[" , i , " , j , "] = " , C[i,j]);
28             fimpara;
29         fimpara;
30     fim.
```

79

GIT - ***Algoritmos***

estrutura de dados: variáveis compostas multidimensionais – matrizes

exercício:

- 1) Desenvolva um algoritmo que dadas duas matrizes A e B, onde A tem 5 linhas e 5 colunas resolva:

$$C = A * (A + B)$$

$$C = B * B$$

- 2) desenvolva um algoritmo que dada mais uma matriz D, onde D tem 5 linhas e 3 colunas, resolva

$$C = A * D$$

80

GIT - Algoritmos***exercícios: 03/07***

- 1) escrever um algoritmo que multiplique duas matrizes do tipo $A_{m,n}$ e $B_{r,s}$ e coloque o resultado numa terceira matriz. Fazer o fluxograma desse algoritmo.
- 2) Dadas duas matrizes continuando o exercício anterior, fixar o m e o s e permitir a variação do n e do r verificando quando é possível fazer adição e multiplicação destas matrizes. Assumir uma matriz C para a resposta com seus tamanhos adequados. Faça um fluxograma mostrando quais as diferenças entre os dois algoritmos e fluxogramas. Logo assuma $C = A + B$ e depois $C = A * B$

81

GIT - Algoritmos

FORBELLONE, ELBERSPÄTCHER, Logica de Programação, 3.a e 4.a Ed. Pearson – Bookman, 202
https://www.ime.usp.br/~leo/mac2166/2017-1/line_introducao_while.html, prof. L. O. Brandão –
 LINE.USP.BR – em 02/07/2024 (fluxogramas para e enquanto)
 MENEZES, Introdução à Programação com Python, 4.a ed, Novatec, 2024

82

Gestão de Tecnologia da Informação

Algoritmos

Prof Carlos Eduardo Câmara (dinho)
Aula 18



83

GIT - Algoritmos

estrutura de dados: variáveis compostas multidimensionais – matrizes

- 1) dada uma matriz A com 5 linhas e 5 colunas, construa um algoritmo que forneça:
 - a) diagonal principal. $a[i, i]$ ou $a[j, j]$ ou seja, quando o índice da linha é igual os índice da coluna
 - b) triângulo superior a diagonal principal índice da linha vai de 1 até o índice do número da coluna, isto é considerando $a[i, j]$, variar i de zero até $i \leq j$
 - c) diagonal secundária
 - d) todos os elementos exceto a diagonal principal

84

GIT - Algoritmos

estrutura de dados: variáveis compostas multidimensionais – matrizes

- 2) desenvolver um algoritmo que dada uma matriz com 6 linhas e 4 colunas:
- a) a soma de cada uma das 6 linhas
 - b) a soma de cada uma das 4 colunas
 - c) escreva uma nova matriz com os elementos pares da matriz e zero nas outras posições
 - d) faça a multiplicação das linhas pares por 2 e das linhas ímpares por 3

85

GIT - Algoritmos

estrutura de dados: variáveis compostas multidimensionais – matrizes

- 3) desenvolver um algoritmo que dada uma matriz A com 5 linhas e 5 colunas, faça:
- trocar a quinta linha pela segunda e escrever a matriz resultante
 - trocar da quarta coluna pela primeira e escrever a matriz resultante
 - trocar a diagonal principal pela diagonal secundária e escrever a matriz resultante

\

86

GIT - Algoritmos***estrutura de dados:***

- 4) desenvolver um algoritmo que leia dois vetores de número, verificar se o vetor está em ordem decrescente. Senão, colocar em ordem crescente.
- somar todos os valores do vetores e calcular a média aritmética.
- usar repetição para ordenar os valores e calcular média aritmética
- escrever o conteúdo do vetor a cada passo na repetição da ordenação
- no final escrever a média aritmética
- fazer o fluxograma desse algoritmo

87

GIT - Algoritmos***estrutura de dados:***

- 5) desenvolver um algoritmo que a partir de matrizes com a configuração:
- A uma matriz com 5 linhas e 5 colunas
- B uma matriz com 5 linhas e 3 colunas
- C uma matriz com 3 linhas e 5 colunas
- D uma matriz com 5 linhas e 5 colunas
- verifique se as seguintes operações são possíveis:
- $A*B$, $A*C$, $A*D$, $B*A$, $B*B$, $C*D$, $C*C$
- escrever como saída se as operações são possíveis e justificar o motivo se for verdadeiro ou falso
- propor uma solução para executar as operações possíveis variando apenas o número de linhas e colunas no início das repetições

88

GIT - Algoritmos***estrutura de dados:***

6) dado o vetor1 [3 8 6 4 7 12 11 9 10 5 2 1]

desenvolver um algoritmo que separe em um vetor com os números impares e outro com os números pares

some os dois vetores

7) dado o vetor [1 2 3 4 5]

usar o algoritmo desenvolvido para contar quantos passos seriam necessários para colocar este vetor em ordem decrescente

8)

89

GIT - Algoritmos***estrutura de dados:*** Exercícios: 04/07

Dados um vetor, duas matrizes e um número inteiro, que devem ser inseridos por um usuário, desenvolver um algoritmo que:

1) leia duas matrizes $A_{m \times n}$ e $B_{r \times m}$

2) leia um vetor c_n

3) mostre ao usuário quais as possíveis relações entre n, r e m para que as operações:

adição, subtração e multiplicação

4) a partir de uma escolha após o item 4, o usuário deve entrar com n, r e m, as matrizes, o vetor

5) verificar se n, r e m são entradas possíveis para o item 6

6) realizar $A*B$, $B*A$, $A*A$, $B*B$, quando possível, justificando a impossibilidade

7) realizar $c*A$, $c*B$, idem ao item anterior

procure desenvolver um código único para realizar as operações aritméticas!

bom trabalho!

90