

Modularização

Comentários

Exemplo de uma matriz inca:

$$M = \begin{array}{|cccc|} \hline 1 & 2 & 3 & 4 \\ \hline 12 & 13 & 14 & 5 \\ \hline 11 & 16 & 15 & 6 \\ \hline 10 & 9 & 8 & 7 \\ \hline \end{array}$$

Modularização

Comentários

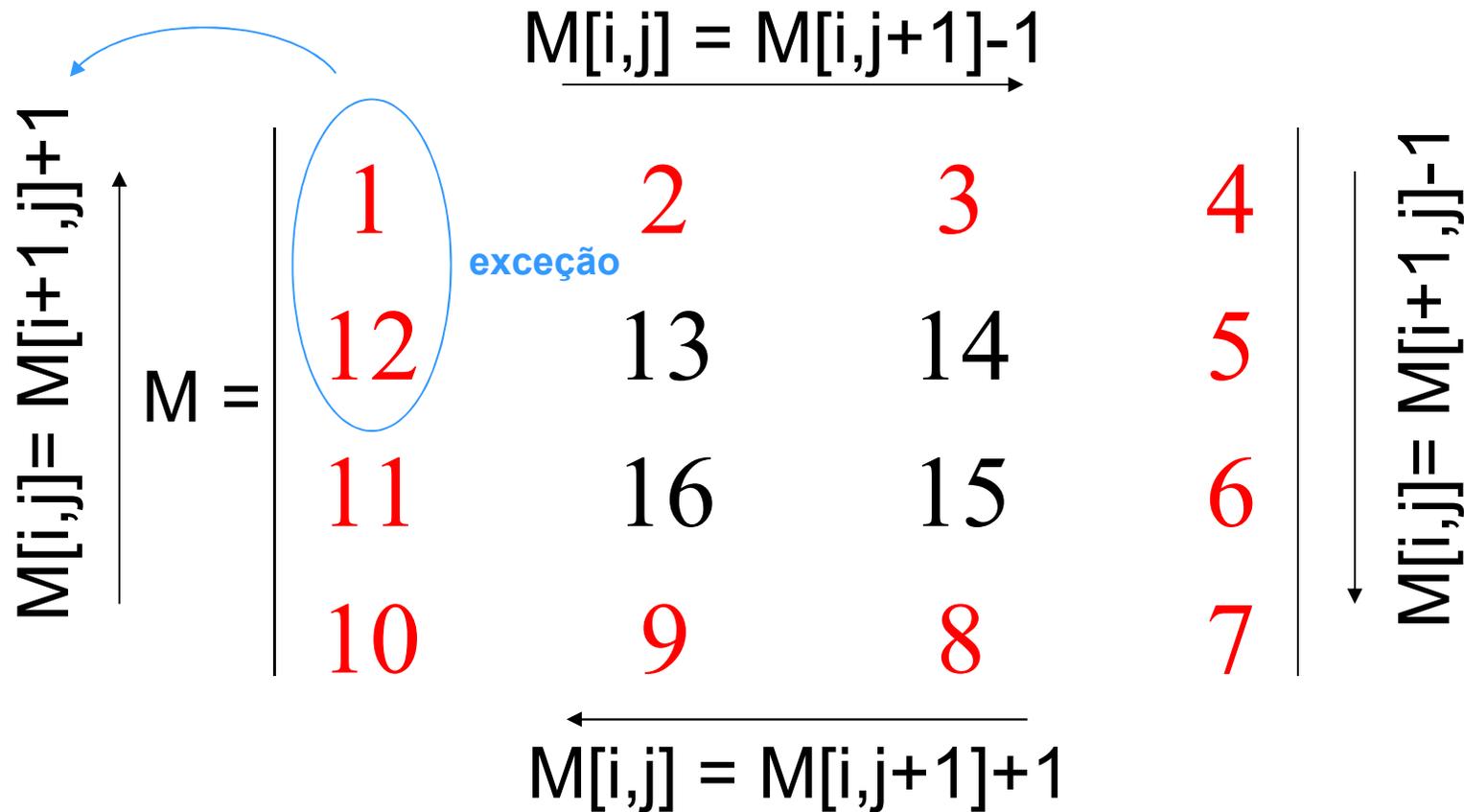
Exemplo de uma matriz inca:

$$M = \begin{array}{|cccc|} \hline 1 & 2 & 3 & 4 \\ \hline 12 & 13 & 14 & 5 \\ \hline 11 & 16 & 15 & 6 \\ \hline 10 & 9 & 8 & 7 \\ \hline \end{array}$$

Modularização

Comentários

Exemplo de uma matriz inca:



Modularização

Comentários

Exemplo de uma matriz inca:

$$M = \begin{array}{|cccc|} \hline 1 & 2 & 3 & 4 \\ \hline 12+1 = 13 & & 14 & 5 \\ \hline 11 & \text{relação} & 16 & 15 \\ \hline 10 & 9 & 8 & 7 \\ \hline \end{array}$$

Modularização

Comentários

Exemplo de uma matriz inca com ordem ímpar:

M =	1	2	3	4	5
	16	17	18	19	6
	15	24	25	20	7
	14	23	22	21	8
	13	12	11	10	9

Obs.: O detalhe é o elemento central mantém a relação $M[o/2+1,o/2+1]-1=M[o/2+1,o/2+1-1]$. “o” é ordem da matriz.

Modularização

Exercício 42: Com o objetivo de melhor fixar alguns aspectos abordados, trabalharemos neste exercício a diferença existente entre o tipo primitivo “caractere” disponível para construção de algoritmos e o tipo primitivo “char” disponível na linguagem C.

Na linguagem C uma string é um vetor de caracteres. Porém, obrigatoriamente um dos caracteres do vetor deve ser o caractere nulo, ou seja, o '\0'. O caractere nulo sucede o último caractere válido da string em questão.

Com base no que foi exposto, construa um algoritmo que manipule um vetor de strings com 10 elementos onde o comprimento máximo das string é de 30 caracteres. As manipulações que devem ser efetuadas são: inicializar vetor, com strings fornecidas pelo usuário; e imprimir o conteúdo do vetor, após a inicialização.

funcao caractere_valido (aux: caractere): logico

inicio

**retorne (aux=" " ou aux="!" ou aux="#" ou aux="\$" ou aux="%" ou aux="&" ou
aux="'" ou aux="(" ou aux=")" ou aux="*" ou aux="+" ou aux="," ou aux="-" ou
aux="." ou aux="/" ou aux="0" ou aux="1" ou aux="2" ou aux="3" ou aux="4" ou
aux="5" ou aux="6" ou aux="7" ou aux="8" ou aux="9" ou aux=":" ou aux=";" ou
aux="<" ou aux="=" ou aux=">" ou aux="?" ou aux="@ " ou aux="A" ou aux="B" ou
aux="C" ou aux="D" ou aux="E" ou aux="F" ou aux="G" ou aux="H" ou aux="I" ou
aux="J" ou aux="K" ou aux="L" ou aux="M" ou aux="N" ou aux="O" ou aux="P" ou
aux="Q" ou aux="R" ou aux="S" ou aux="T" ou aux="U" ou aux="V" ou aux="W" ou
aux="X" ou aux="Y" ou aux="Z" ou aux="[" ou aux="\" ou aux="]" ou aux="^" ou
aux="_" ou aux="`" ou aux="{ " ou aux="|" ou aux="}" ou aux="~")**

fimfuncao

algoritmo "vet_strings"

var

vet_strings: vetor [1..10, 1..31] de caractere

i, j: inteiro

aux: caractere

funcao caractere_valido (aux: caractere): logico

inicio

retorne (...)

fimfuncao

Inicio

para i de 1 ate 10 faca

escreva ("Obs.: Digite um a um os caracteres que compõem a")

escreval (" string, pressionando após")

escreval ("cada caractere a tecla enter.")

escreva ("A leitura de uma string é finalizada pelo fornecimento")

escreval (" de 30 caracteres ou")

escreval ("pelo fornecimento do caractere espaço ' '.")

escreval ("Entre com os caracteres da ",i,"ª string:")

para j de 1 ate 30 faca

repita

leia (aux)

ate (caractere_valido(aux))

se (aux=" ") então

interrompa

senao

vet_strings[i,j]<-aux

fimse

301 fimpara

```

vet_strings[i,j]<-" 0"
se (j=31) entao
  escreval ("Ateno! Tamanho mximo da ",i,"a string atingido.")
  se (i<>10) entao
    escreval ("Os prximos caracteres a serem digitados pertencero
a ",i+1,"a string")
  fimse
fimse
fimpara
escreval ("Contedo do vetor.")
para i de 1 ate 10 faca
  escreva (i,"a string: ")
  para j de 1 ate 30 faca
    se (vet_strings[i,j]<>" 0") entao
      escreva(vet_strings[i,j])
    senao
      interrompa
  fimse
fimpara
escreval("")
fimpara
fimalgoritmo
//Obs.: nesta resposta no foi explorado adequadamente o conceito de
//modularizao, pois a inteno foi deixar o algoritmo compatvel com
//o VisuAlg. Possibilitando que os alunos o utilizem para interpretar
//a soluo.

```

Modularização

Exercício 42-b:

Considerando a movimentação de uma rainha no tabuleiro de xadrez, desenvolveremos um módulo capaz de identificar se um determinado tabuleiro ocupado por apenas 8 rainhas encontra-se em uma configuração válida para o jogo de oito rainhas.

Uma configuração válida para o tabuleiro no jogo de oito rainhas, consiste em um tabuleiro onde cada uma das oito rainhas está posicionada em uma casa onde nenhuma das demais pode atingi-la com apenas um movimento.

Com base no que foi exposto, construa um módulo que receba uma matriz 8×8 de inteiros e retorne um lógico correspondendo ao fato da matriz recebida representar ou não um tabuleiro com uma configuração válida para o jogo de oito rainhas.

Observação: considere que casas desocupadas no tabuleiro conterão o valor 0 (zero) e casas ocupadas por rainhas terão o valor 1 (um).

Modularização

