

TODOS OS TESTES DE MESA NA PROVA

1) Qual o resultado armazenado em vetC, após a execução do algoritmo? (1.0)

```
início
constante N 6;
inteiro vetA[N] ← {10, 40, 80, 10, 30, 50};
inteiro vetB[N] ← {20, 30, 80, 20, 20, 50};
inteiro vetC[N];
inteiro i;
para i de N - 1 até 0 passo -1
  faça
    se (vetA[i] < vetB[i])
      então vetC[N-i-1] ← vetB[i] - vetA[i];
      senão vetC[N-i-1] ← vetA[i] - vetB[i];
    fimse
  fimpara
para i de 0 até N-1
  faça escreva(vetC[i]);
fimpara
fim
```

Teste de Mesa:

i N vetC

5	6	[0	-> 50 = 50 ----- vetC(0) <- 50 - 50 -> vetC(0) = 0
4	5	[0, 10	-> 30 < 20 senão vetC(1) <- 30 - 20 -> vetC(1) = 10
3	4	[0, 10, 10	-> 10 < 20 então vetC(2) <- 20 - 10 -> vetC(2) = 10
2	3	[0, 10, 10, 0	-> 80 = 80 ----- vetC(3) <- 80 - 80 -> vetC(3) = 0
1	2	[0, 10, 10, 0, 10	-> 40 < 30 senão vetC(4) <- 40 - 30 -> vetC(4) = 10
0	1	[0, 10, 10, 0, 10, 10	-> 10 < 20 então vetC(5) <- 20 - 10 -> vetC(5) = 10

Resp: vetC =

0	10	10	0	10	10
---	----	----	---	----	----

Observação: Note que os vetores vetA e vetB estão sendo criados e inicializados, assim evitamos o processo de leitura. Isto também é possível na linguagem C.

2) Dada a matriz MAT declarada abaixo e já lida. (Valor

1.0)

MAT: vetor[0..3, 0..3] de caractere;

O	Q	*	T
E	A	E	S
R	E	U	T
A	*	*	S

Qual será a sua configuração depois de executados os comandos:

```
para I de 0 até 3 passo 1
  faça para J de I+1 até 3 passo 1
    faça AUX MAT[I][J];
    MAT[I][J] MAT[J][I];
    MAT[J][I] AUX;
  fimpara;
fimpara;
AUX MAT[0][0];
MAT[0][0] MAT[3][3];
MAT[3][3]= AUX;
AUX MAT[1][1];
MAT[1][1] MAT[2][2];
MAT[2][2] AUX;
```

Teste de Mesa:

MAT	0	1	2	3	i	j	aux
0	s	e	r	a	0	-	q
1	q	u	e	t	0	1	*
2	*	e	a	*	0	2	t
3	s	*	t	o	0	3	e
					1	2	e
					1	3	s
					2	3	t
					3	-	o
							a

Resposta:

S	E	R	A
Q	U	E	T
*	E	A	*
S	*	T	O

3) Qual a saída do algoritmo? Considere como entrada os valores: 12, 30, 40, 2, 5, 8, 9, nesta ordem. (valor 1.0)

```
algoritmo "Exercício3"
var
  vet:vetor[1..7] de inteiro
  x, y,i:inteiro
inicio
  escreval("Entre com 7 números inteiros:")
  para i de 1 ate 7 faca
    escreva("Entre com o", i,"º número:")
    leia(vet[i])
  fimpara
  x<-vet[1]
  y<-1
  para i de 2 ate 7 faca
    se (x > vet[i]) entao
      x <- vet[i]
    y <-i
  fimse
  fimpara
  escreva(x, y)
finalgoritmo
```

Teste de mesa:

I	VET	X	Y
		12	1
2	30		
3	40		
4	2	2	4
5	5		
6	8		
7	9		

Resposta: 2, 4

4) Faça um programa em C que leia uma matriz NxM de valores reais, encontre o maior valor de cada linha e o armazene em um vetor. Encontre também o maior valor da matriz e utilize para dividir todos os elementos da matriz por ele. (6.0)

Então apresente nesta ordem:

- a matriz lida
- o vetor com o maior valor de cada linha
- o maior valor da matriz
- a matriz dividida pelo maior valor encontrado
- a soma dos elementos da matriz após a divisão
- a transposta da matriz depois da divisão

Parte Prática

5) Interprete o pseudo-código abaixo e preencha o quadro com a matriz e vetor resultantes da execução: (1.0)

início

inteiro inteiro inteiro L, C;

inteiro inteiro inteiro matriz[4][4], vetor[4];

para L de 0 até 3 passo 1 faça

para C de 0 até 3 passo 1 faça

se se L = C então

matriz[L][C] L;

vetor[L] (L + C) * 2;

senão

se $L > C$ então

matriz[L][C] $L - C$;

senão

matriz[L][C] $C - L$;

fim-se

fim- fim-se

fim- fim-para

fim- fim-para

fim

Teste de mesa:

0	0	0	0
0	1	1	
0	2	2	
0	3	3	
1	0	1	
1	1	1	4
1	2	1	
1	3	2	
2	0	2	
2	1	1	
2	2	2	8
2	3	1	
3	0	3	
3	1	2	
3	2	1	
3	3	3	12

Matriz resultante:

0	1	2	3
1	1	1	2
2	1	2	1
3	2	1	3

Vetor resultante:

0	4	8	12
----------	----------	----------	-----------