

**Questão 1 )** (1 ponto) Considere que os vetores declarados abaixo contenham os valores indicados a seguir:

var

X, Y, Z: vetor[1..5] de inteiro;

X	1	9	6	4	10
Y	2	1	8	4	5
Z	2	9	8	4	10

Após a execução do trecho de código abaixo, indique os valores armazenados no vetor Z:

```

para i <- 1 ate 5 do
  inicio
    se (X[i] > Y[i])
      entao Z[i] <- X[i]
    senao Z[i] <- Y[i];
    fimse
  fimpara;

```

- ☒ a) 2; 9; 8; 4; 10 .
- b) 2; 9; 8; indeterminado; 5
- c) 2; 1; 6; 4; 10
- d) 1; 9; 6; 4; 5
- e) 1; 1; 8; indeterminado; 10

**Questão 2 )** (1 ponto) Qual será o resultado final do programa abaixo:

programa Parametros;

var inteiro X,Y,Z,W;

procedimento inicializa ( var A,B,D: inteiro; C:inteiro);

```

inicio
  escreva ('Passo 1:', A, B, C, D);
  A<-B; B<-C; C<-D; D<-1;
  escreva ('Passo 2:', A, B, C, D);
fimprocedimento;

```

```

inicio
  X<-1; Y<-2; Z<-3; W<-4;
  inicializa (X, Y, Z, W);
  escreva ('Passo 3:', X, Y, Z, W);
fimprograma.

```

VAR

X	Y	Z	W
1	2	3	4
A	B	D	C
1	2	4	3

↓

- ☒ a) Passo 1: 1243 // Passo 2: 2431 // Passo 3: 2414
- ☒ b) Passo 1: 1243 // Passo 2: 2431 // Passo 3: 1243
- c) Passo 1: 1243 // Passo 2: 2431 // Passo 3: 2431
- d) Passo 1: 1243 // Passo 2: 1243 // Passo 3: 1243
- e) Passo 1: 1243 // Passo 2: 2414 // Passo 3: 2414

**Questão 3 )** (1 ponto) Considere uma matriz quadrada A, de dimensão 3, contendo os valores indicados abaixo:

1	2	3
1	2	3
1	2	3

Para que o código a seguir armazene tais valores na matriz, devemos substituir a linha contendo o texto **<comandos>** por:

---

```
var
    A: vetor[1..3,1..3] de inteiro;
    i: inteiro;
    ...
    para i <- 1 to 3 do
        inicio
            <comandos>
        fim;
```

---

- ☒ a) A[i,1] <- 1; A[i,2] <- 2; A[i,3] <- 3;
- b) A[1,i] <- 1; A[2,i] <- 2; A[3,i] <- 3;
- c) A[i,i] <- 1; A[i,i] <- 2; A[i,i] <- 3;
- d) A[i,3] <- 1; A[i,2] <- 2; A[i,1] <- 3;
- e) A[i,i] <- i;

**Questão 4 )** (1 ponto) Considerando que o usuário irá informar um valor inteiro para i, e que se deseja elaborar uma função que utilize a técnica de recursividade para informar a soma de 1 até i, conforme segue:

**SOMA <- 1 + 2 + 3 + 4 + ... + i**

Assinale a alternativa que possui que completa, corretamente, o programa abaixo:

```
programa L2Q1;
var
    i: inteiro;
funcao soma(n: inteiro):inteiro;
inicio
    se n = 0 entao
        soma <- 0
    senao
        ..... (assinale a instrução corretamente abaixo)
    fimfuncao;
inicio
    escreva('Entre com o valor de i: '); 10
    leia(i) 10
    escreva('SOMA : ',soma(i));
fimprograma.      soma(n-1) // soma (10-1)
```

- a) soma <- n -1;
- b) soma <- n + soma(1);
- c) soma <- n + soma(n);
- d) soma <- n + soma;
- ☒ e) soma <- n + soma(n-1);



**Questão 5 )** (1 ponto) Matriz **NULA** é uma matriz que possui todos os seus elementos iguais a ZERO.

Considere a matriz quadrada **A**, de dimensão 3, definida abaixo. Para que o código a seguir gere a matriz **A NULA**, devemos substituir a linha contendo o texto **<comando>** por:

```
var
  A: vetor[1..3,1..3] of inteiro;
  i, j: inteiro;
...
para i <- 1 to 3 do
  para j <- 0 to 2 do
    <comando>
```

- ☒ a)  $A[i,j+1] <- 0;$
- b)  $A[i+1,j+1] <- 0;$
- c)  $A[i+1,j] <- 0;$
- d)  $A[i,j] <- 0;$
- e)  $A[i,i] <- 0;$

**Questão 6 )** (1 ponto) Considere o seguinte trecho de pseudocódigo:

```
S: VETOR[1..5] de inteiro;
```

```
i, j: inteiro;
```

```
Para i de 1 até 5 faça
```

```
  S[i] <- 0;
```

```
  Para j de 1 até i faça
```

```
    S[i] <- S[i] + 2*j - 1;
```

$$\begin{aligned} S[1] &= 0 + 2 \cdot 1 - 1 = 1 \\ S[2] &= 1 + 2 \cdot 2 - 1 = 4 \\ S[3] &= 4 + 2 \cdot 3 - 1 = 9 \\ S[4] &= 9 + 2 \cdot 4 - 1 = 16 \\ S[5] &= 16 + 2 \cdot 5 - 1 = 25 \end{aligned}$$

Indique os valores armazenados no vetor S, após a execução do código acima:

- ☒ A) 1; 4; 9; 16; 25
- B) 1; 3; 6; 10; 15
- C) 1; 2; 3; 4; 5
- D) 2; 4; 6; 8; 10
- E) 1; 5; 10; 15; 20

Questão 7 ) (2 pontos) Faça uma rotina que leia uma matriz 6 x 6 e atribua o valor 0 para os valores negativos encontrados na da diagonal principal.

Questão 8 ) (2 pontos) Faça um algoritmo que leia 20 números inteiros e os guarda em 2 vetores: um com os números ímpares e o outro com os números pares. Imprima os 2 vetores.

ATENÇÃO AOS INDICES! Os vetores não poderão ter posições em branco entre os elementos.

PROGRAM Q7

VAR MAT: ARRAY [1..6], [1..6] of INTEGER;

I, J: INTEGER;

BEGIN

WRITELN ('CARREGANDO MATRIZ 6x6');

FOR i := 1 to 6 do

FOR j := 1 to 6 do

BEGIN

READLN (MAT [i, j]);

END;

BEGIN

IF ~~(i=j)~~ <sup>and MAT[i,j]</sup> < 0 Then

MAT [i, j] := 0;

END;

END.

1,5

PROGRAM Q8

VAR VET, VETP, VETI: ARRAY [1..20] of INTEGER;

NUM, I, J: INTEGER;

BEGIN

WRITELN ('CARREGANDO VETOR 20 ELEM.');

FOR i := 1 to 20 do

READ (~~VET [i]~~); (NUM);

BEGIN

IF (i MOD 2 = 0) THEN

VETP [i] := ~~VET [i]~~ NUM

ELSE

VETI [i] := ~~VET [i]~~ NUM

END;

WRITELN ('VETOR PAR:', VETP [i]);

WRITELN (' ');

WRITELN ('VETOR IMPAR:', VETI [i]);

END;

1,0



PROGRAM Q7

VAR MAT: ARRAY [1..6], [1..6] of INTEGER;  
i, j: INTEGER;

BEGIN

WRITELN('CARREGANDO MATRIZ 6x6:');

FOR i: 1 to 6 do

FOR j: 1 to 6 do

BEGIN

READ('MAT[i,j]:');

END;

BEGIN

IF (i=j) THEN

END;

BEGIN

IF

	1	2	3	4	5	6
1	1	2	3	4	5	6
2	4	2				
3	1		3			
4	1			4		
5	4				5	
6	1					6

PROGRAM Q8

VAR VETVETR, VETI: ARRAY [1..20] of  
INTEGER;

i, j, k: INTEGER;

BEGIN

WRITELN('CARREGANDO VETOR 20:')

FOR i: 1 to 20 do

READ('VET[i]:');

BEGIN

IF i MOD 2 = 0 THEN

VETP ← VET

ELSE

VETI ← VET

END;

WRITELN('VETOR PAR:', VETP);

WRITELN('VETOR IMPAR:', VETI);

END.