



Programador de Sistemas

Lógica de Programação

Vetores – 2ª parte

Prof. Mauricio Wieler Orellana

mauricioow@gmail.com

1. Crie um algoritmo que leia um vetor com 8 números inteiros, leia um número X e multiplique o vetor pelo número X lido e mostre o vetor anterior e o vetor multiplicado pelo número X

Ex: Vetor VET

0	1	2	3	4	5	6	7
5	8	2	25	9	4	13	7

X = -2 => o vetor será:

-10	-16	-4	-50	-18	-8	-26	-14
-----	-----	----	-----	-----	----	-----	-----

Índices do vetor
VET
(i)

Elementos do vetor
VET

VET[i]

Exemplo:

VET[0] = 5

VET[1] = 8

...

VET[2] = 13

VET[3] = 7

3º

1º

2º

```
algoritmo "Vetor-04-07_1"
var
    V,W:vetor[1..8] de inteiro
    i,x:inteiro
inicio
    //Rotina de leitura
    para i de 1 ate 8 faca
        escreval("Digite V[,i,]:")
        leia(V[i])
    fimpara
    escreval("Digite um valor constante inteiro:")
    leia(x)
    //Rotina de Impressão do vetor V
    escreval
    escreval("Vetor V (original):")
    para i de 1 ate 8 faca
        escreva(V[i]," ")
        W[i] <- V[i]*x
    fimpara
    escreval
    escreval("Vetor W (multiplicado pela constante [,x,]):")
    para i de 1 ate 8 faca
        escreva(W[i]," ")
    fimpara
finalgoritmo
```

2. Escrever um algoritmo que peça para que o usuário leia dois vetores A e B de 6 elementos cada um e crie e mostre na tela o vetor SOMA e o vetor DIF, que serão, respectivamente, a soma e a diferença, dos elementos correspondentes a A e B:

Exemplo:

A	0	1	2	3	4	5
	3	8	15	-9	6	4
	0	1	2	3	4	5
	2	4	-13	50	5	8
	0	1	2	3	4	5
	5	12	2	41	11	12

B	0	1	2	3	4	5
	2	4	-13	50	5	8
	0	1	2	3	4	5
	2	4	-13	50	5	8
	0	1	2	3	4	5
	2	4	-13	50	5	8

SOMA	0	1	2	3	4	5
	5	12	2	41	11	12
	0	1	2	3	4	5
	2	4	-13	50	5	8
	0	1	2	3	4	5
	2	4	-13	50	5	8

DIF	0	1	2	3	4	5
	1	4	28	-51	1	-4
	0	1	2	3	4	5
	2	4	-13	50	5	8
	0	1	2	3	4	5
	2	4	-13	50	5	8

Diagram illustrating the calculation of the sum (SOMA) and difference (DIF) of two vectors A and B. The vectors A and B are shown with their elements. The SOMA vector is calculated as A + B, and the DIF vector is calculated as A - B. The steps are labeled: 1º. passo (Step 1), 2º. passo (Step 2), and 3º. passo (Step 3).

```
Para i de 0 ate 5 faça  
    SOMA[i] <- A[i] + B[i]  
    DIF[i] <- A[i] - B[i] //Diferença é a SUBTRAÇÃO  
fimpara
```

Repetição que calcula a soma e a subtração dos elementos dos vetores A e B

```
algoritmo "Vetor Soma e Vetor Diferença"
var
    A, B, SOMA, DIF:vetor[1..8] de inteiro
    i:inteiro
inicio
    //Rotina de leitura
    escreval("Carregando o vetor A:")
    para i de 1 ate 8 faca
        escreval("Digite A[",i,"]:")
        leia(A[i])
    fimpara
    escreval("Carregando o vetor B:")
    para i de 1 ate 8 faca
        escreval("Digite B[",i,"]:")
        leia(B[i])
        SOMA[i] <- A[i] + B[i]
        DIF[i] <- A[i] - B[i]
    fimpara
    escreval("Vetor Soma: ")
    para i de 1 ate 8 faca
        escreva(SOMA[i]," ")
    fimpara
    escreval
    escreval("Vetor Diferença: ")
    para i de 1 ate 8 faca
        escreva(DIF[i]," ")
    fimpara
fimalgoritmo
```

3. Escreva um algoritmo que peça para o usuário digitar os valores de um vetor de inteiros com 10 elementos e em seguida exiba o valor da soma dos elementos do vetor e o menor elemento do vetor e sua posição.

	1	2	3	4	5	6	7	8	9	10
VECT	3	5	8	-1	14	-7	22	5	6	1
	0	1	2	3	4	5	6	7	8	9

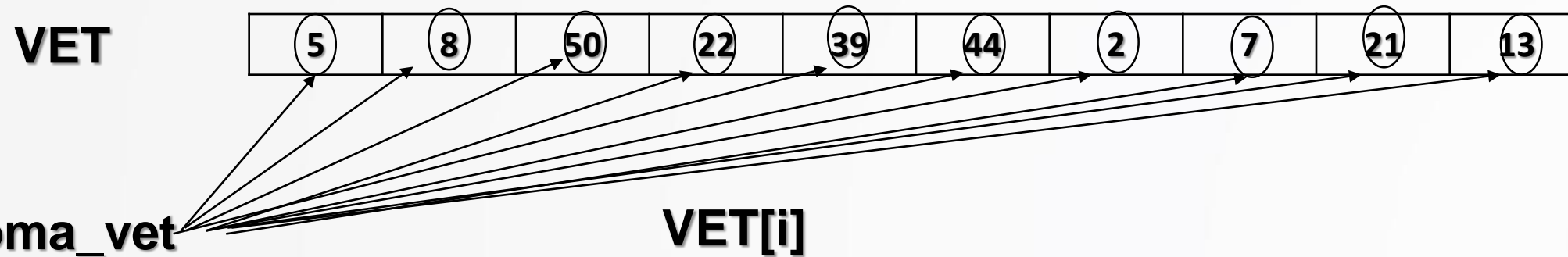
$$\text{Soma} = 3 + 5 + 8 + (-1) + 14 + (-7) + 22 + 5 + 6 + 1 = 56$$

Menor n°: -7 ; posição do menor n°: 6

1º.

2º.

3º.



$$\text{soma_vet} = 5 + 8 + 50 + 22 + 39 + 44 + 2 + 7 + 21 + 13 = 211$$

```
para i de 1 ate 10 faca  
    soma_vet <- soma_vet + VET[i]  
fimpara
```

```
algoritmo "Vetor com 10 elementos - Soma dos elementos e menor
elemento"
var
    VET:vetor[1..10] de inteiro
    i,menor,soma,posicao:inteiro
inicio
    //Rotina de leitura
    soma <- 0
    escreval("Carregando o vetor VET:")
    para i de 1 ate 10 faca
        escreval("Digite VET[",i,"]:")
        leia(VET[i])
        soma <- soma + VET[i]
    fimpara
    menor <- VET[1]
    posicao <- 1
    //Início das Comparações
    para i de 2 ate 10 faca
        se (VET[i] < menor) entao
            menor <- VET[i]
            posicao <- i
        fimse
    fimpara
    escreval
    escreval("A soma dos elementos do vetor, vale: ",soma)
    escreval("O menor número digitado no vetor é: ",menor)
    escreval("e está localizado na posição: ",posicao)
finalgoritmo
```


4. Crie um algoritmo que leia um vetor com 8 números inteiros, calcule e mostre na tela os números primos e suas respectivas posições.

Ex:

5	8	2	25	9	4	13	7
---	---	---	----	---	---	----	---

Deve mostrar na tela:

Os números primos são:

5 na posição 1

2 na posição 3

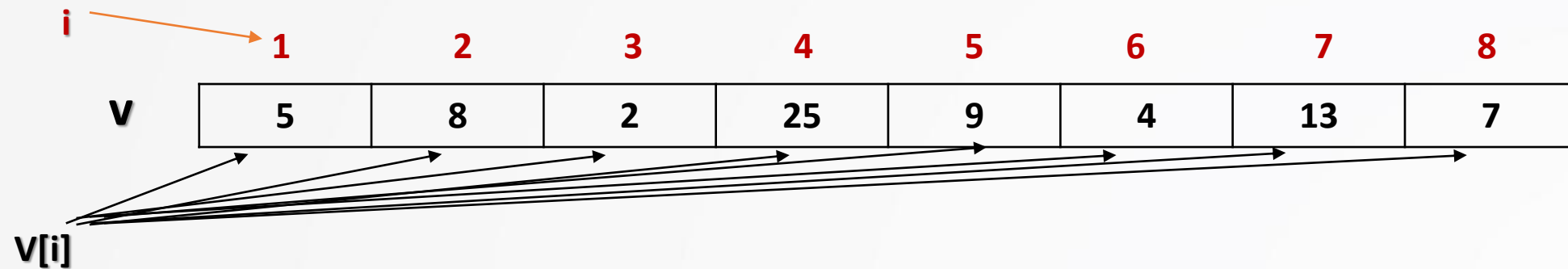
13 na posição 7

7 na posição 8

2º

1º

3º



divisor	i	V[i]	V[i] % divisor	cont_divisor
1	1	5	$5 \% 1 = 0$	0
2			$5 \% 2 = 1$	$0 + 1 = 1$
3			$5 \% 3 = 2$	$1 + 1 = 2$
4			$5 \% 4 = 1$	
5			$5 \% 5 = 0$	

NA TELA:

5 na posição 1 (Só testamos o primeiro elemento do vetor, este teste vai até $v[8] = 7$)

2 na posição 3

13 na posição 7

7 na posição 8

```
algoritmo "Números Primos no vetor"
var
    NUM:vetor[1..8] de inteiro
    i,j,cont:inteiro
inicio
    //Rotina de leitura
    escreval("Carregando o vetor NUM:")
    para i de 1 ate 8 faca
        escreval("Digite NUM[",i,"]:")
        leia(NUM[i])
    fimpara
    limpatela
    escreval("Números Primos no Vetor: ")
    para i de 1 ate 8 faca
        cont <- 0
        para j de 1 ate NUM[i] faca
            se (NUM[i]%j = 0) entao
                cont <- cont + 1
            fimse
        fimpara
        se (cont = 2) entao
            escreval(NUM[i]," na posição ",i)
        fimse
    fimpara
fimalgoritmo
```

5. Crie um algoritmo que leia e carregue um vetor de 10 posições, calcule e mostre:

- O maior elemento do vetor e em que posição esse elemento está.

- O menor elemento do vetor e em que posição esse elemento está.

1º.

3º.

2º.

```

algoritmo "Maior e menor no vetor"
var
    VET:vetor[1..10] de inteiro
    i,maior,menor,posmaior,posmenor:inteiro
inicio
    //Rotina de leitura
    escreval("Carregando o vetor VET:")
    para i de 1 ate 10 faca
        escreval("Digite VET[",i,"]:")
        leia(VET[i])
    fimpara
    maior <- VET[1]
    menor <- VET[1]
    posmaior <- 1
    posmenor <- 1
    limpatela
    //Início das Comparações
    para i de 2 ate 10 faca
        se (VET[i] < menor) entao
            menor <- VET[i]
            posmenor <- i
        fimse
    fimpara
    para i de 2 ate 10 faca
        se (VET[i] > maior) entao
            maior <- VET[i]
            posmaior <- i
        fimse
    fimpara
    escreval
    escreval("O maior número digitado no vetor é: ",maior)
    escreval("e está localizado na posição: ",posmaior)
    escreval("O menor número digitado no vetor é: ",menor)
    escreval("e está localizado na posição: ",posmenor)
fimalgoritmo

```

Sugestão:

TESTE DE MESA

Carregar o vetor com valores quaisquer, no papel e realizar o teste de mesa

6. Crie um vetor com 5 elementos e calcule o fatorial de cada elemento do vetor:

V

3	5	4	2	7
---	---	---	---	---

FAT

6	120	24	2	5040
---	-----	----	---	------

Handwritten annotations: Green arrows point from each element in the top row to its factorial in the bottom row. The arrows are labeled with the factorial symbol and the number: $3!$, $5!$, $4!$, $2!$, and $7!$. A large green arrow points from the $5!$ calculation to the $4!$ calculation, indicating a relationship between the two.


```
algoritmo "Fatoriais no vetor"
var
    NUM:vetor[1..8] de inteiro
    i,j:inteiro
    fat:real
inicio
    //Rotina de leitura
    para i de 1 ate 8 faca
        escreval("Digite V[,i,]:")
        leia(NUM[i])
    fimpara
    limpatela
    escreval("Vetor original: ")
    para i de 1 ate 8 faca
        escreva(NUM[i], " ")
    fimpara
    escreval
    escreval
    escreval("Vetor de Fatoriais: ")
    para i de 1 ate 8 faca
        fat <- 1
        para j de 1 ate NUM[i] faca
            fat <- fat * j
        fimpara
        escreval(fat)
    fimpara
fimalgoritmo
```

7. Crie um algoritmo que leia dois vetores A e B de 8 posições e crie um vetor INTER, intercalado, com os elementos de A e B.

1	2	3	4	5	6	7	8
10	8	4	6	20	18	34	-2

1	2	3	4	5	6	7	8
15	19	24	1	-1	0	-5	3

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
10	15	8	19	4	24	6	1	20	-1	18	0	34	-5	-2	3

$$\text{INTER}[1] = A[1] = 10 \quad \text{INTER}[2] = B[1] = 15$$

$$\text{INTER}[3] = A[2] = 8 \quad \text{INTER}[4] = B[2] = 19$$

$$\text{INTER}[5] = A[3] = 4$$

$$\text{INTER}[6] = B[3] = 24$$

1º

2º

A

2	5	8	6	3	9	120	1
---	---	---	---	---	---	-----	---

B

6	4	1	20	33	19	5	0
---	---	---	----	----	----	---	---

INTER

2	6	5	4	8	1	6	20	3	33	9	19	120	5	1	0
---	---	---	---	---	---	---	----	---	----	---	----	-----	---	---	---

Senac

```

algoritmo "Vetor Intercalado"
var
    A, B:vetor[1..8] de inteiro
    INTER: vetor[1..16] de inteiro
    i, cont:inteiro
inicio
    //Rotina de leitura
    escreval("Carregando o Vetor A:")
    para i de 1 ate 8 faca
        escreval("Digite A[",i,"]:")
        leia(A[i])
    fimpara
    limpatela
    escreval("Carregando o Vetor B:")
    para i de 1 ate 8 faca
        escreval("Digite B[",i,"]:")
        leia(B[i])
        INTER[2 * i - 1] <- A[i]
        INTER[2 * i] <- B[i]
    fimpara
    limpatela
    escreval("Vetor Intercalado: ")
    para i de 1 ate 16 faca
        escreva(INTER[i], " ")
    fimpara
    escreval
finalgoritmo

```

```
algoritmo "Vetor - Intercalação de 2 vetores de 8 elementos inteiros"
var
  A,B: vetor[1..8] de inteiro
  INTER: vetor[1..16] de inteiro
  i: inteiro
inicio
  escreval("Vetor - Intercalação de 2 vetores de 8 elementos inteiros")
  //1o. passo - Leitura dos vetores A e B de 8 inteiros
  escreval("Carregando o vetor A de 8 inteiros: ")
  escreval("-----")
  para i de 1 ate 8 faca
    escreval("Digite A[",i,"]: ")
    leia(A[i])
  fimpara
  escreval
  escreval("Carregando o vetor B de 8 inteiros: ")
  escreval("-----")
  para i de 1 ate 8 faca
    escreval("Digite B[",i,"]: ")
    leia(B[i])
    INTER[2*i - 1] <- A[i]
    INTER[2*i] <- B[i]
  fimpara
  limpatela
  escreval("Impressão do 1o. vetor - Vetor A: ")
  escreval("-----")
```

Escopo	Nome	Tipo	Valor
GLOBAL	A[1]	I	5

```
escreval("Impressão do 1o. vetor - Vetor A: ")
escreval("-----")
para i de 1 ate 8 faca
    escreva(A[i], " ")
fimpara
escreval
escreval("Impressão do 2o. vetor - Vetor B: ")
escreval("-----")
para i de 1 ate 8 faca
    escreva(B[i], " ")
fimpara
escreval
escreval("Impressão do 2o. vetor - Vetor INTERCALADO: ")
escreval("-----")
para i de 1 ate 16 faca
    escreva(INTER[i], " ")
fimpara
finalgoritmo
```

Escopo	Nome	Tipo	Valor
GLOBAL	A[1]	I	5

39:22 Modificado Use Ctrl+J para acessar a lista de comandos e funções do Visualg 2.5



Digite aqui para pesquisar

ENG
INTL11:20
13/11/2020

8. Crie um algoritmo que leia um vetor de dez números inteiros e calcule e mostre dois vetores resultantes:

- Um vetor com os números pares
- Um vetor com os números ímpares

V

14	3	8	5	16	4	13	15	21	24
----	---	---	---	----	---	----	----	----	----

P

14	8	16	4	24
----	---	----	---	----

I

3	5	13	15	21
---	---	----	----	----

```

algoritmo "Vetor de Pares e vetor de ímpares"
var
    V, P, IMP:vetor[1..10] de inteiro
    i, contp, conti: inteiro
inicio
    //Rotina de leitura
    escreval("Carregando um vetor com 10 elementos inteiros: ")
    contp <- 0
    conti <- 0
    para i de 1 ate 10 faca
        escreval("Digite V[",i,"]:")
        leia(V[i])
        se V[i]%2 = 0 entao
            contp <- contp + 1
            P[contp] <- V[i]
        senao
            conti <- conti + 1
            IMP[conti] <- V[i]
        fimse
    fimpara
    limpatela
    se (contp = 0) entao
        escreval("Não há pares no vetor")
    senao
        escreval("Vetor de Pares: ")
        para i de 1 ate contp faca
            escreval(P[i])
        fimpara
    fimse
    se (conti = 0) entao
        escreval("Não há ímpares no vetor")
    senao
        escreval("Vetor de Ímpares: ")
        para i de 1 ate conti faca
            escreval(IMP[i])
        fimpara
    fimse
fimalgoritmo

```