

Arranjos

Roberto Rocha



Programar é simples?

Você está seguindo nossas regras?

Esta estudando?



Programar

Segundo Carlos Almeida*, professor da Escola Secundaria Emídio Navarro, da região de Viseu, em Portugal,

“Algoritmo não é a solução de um problema, pois, se assim fosse, cada problema teria um único algoritmo”,

“Algoritmo é um caminho para a solução de um problema, em geral, os caminhos para uma solução são muitos”

O desenvolvimento de algoritmos utiliza procedimentos lógicos e de raciocínio na busca da solução de problemas.

Um mesmo problema pode ser resolvido de varias formas, tanto por uma pessoa como por um conjunto de pessoas.

Resolver um problema de varias formas não significa, em absoluto, escrever a solução de qualquer jeito.

A escrita de algoritmos deve respeitar um certo formalismo. A base do desenvolvimento de algoritmos, do ponto de vista computacional, é o próprio computador.

Programar

É necessário ao programador ter seu pensar modelado no computador, e para pensar "computador" é preciso seguir as regras que estamos realizando.

Programar não é trabalho simples.

Se programar um computador fosse simples, todas as pessoas o fariam e não seria necessário muito esforço para o aprendizado.

A grande dificuldade está exatamente no fato de existirem vários caminhos para a solução de um problema e ter de seguir regras de trabalho.

Como saber o caminho correto?

Qual é o melhor algoritmo para o problema?

Estude!!!

ESTUDE!!!

Pratique!!!

PRATIQUE!!!

Programar

O professor Carlos Almeida* adverte que o aprendizado de algoritmos não se faz copiando algoritmos, mas construindo e testando-os, ou seja, exercitando algoritmos.

Não há outra forma de fazê-lo, pois "o único lugar onde o sucesso vem antes do trabalho é no dicionário", como disse Albert Einstein.

Por mais experiência que tenha um programador, em algum momento da vida profissional terá de elaborar e testar algoritmos muitas vezes complexos e diferentes da forma que esteja acostumado.

O programador é um eterno estudante.

A construção de algoritmos voltados à programação de computadores exige cuidado e atenção.

Assim que a solução de um certo problema computacional, por meio de um algoritmo, é elaborada, é necessário realizar testes para verificar se a linha de raciocínio é correta.

Exercício

Faça um algoritmo para ler três notas de alunos e imprimir quantas notas tem valor abaixo da média

Exercício

Faça um algoritmo para ler três notas e imprimir quantas notas tem valor abaixo da média

```
7 Var
8 // Seção de Declarações das variáveis
9 n1,n2,n3,media:real
10 nr:inteiro
11 Inicio
12 // Seção de Comandos
13 leia(n1,n2,n3)
14 media<-(n1+n2+n3)/3
15 nr<-0
16 se n1<media entao
17     nr<-nr+1
18 fimse
19 se n2<media entao
20     nr<-nr+1
21 fimse
22 se n3<media entao
23     nr<-nr+1
24 fimse
25 escreva(nr)
26 Fimalgoritmo
```

Imagina se fosse um programa para 60 alunos?
Como ficaria o programa?

Estrutura de dados homogênea

Vamos descrever uma técnica de programação que auxilia o agrupamento de dados do mesmo tipo em uma mesma variável, porém indexada.

Devido a essa característica referencia-se essa técnica como estrutura de dados homogênea (mesmo tipo de dado).

A estrutura de dados homogênea em programação recebe diversos nomes, como:

variáveis indexadas,

variáveis compostas,

variáveis subscritas,

arranjos,

vetores,

matrizes,

tabelas em memória,

arrays (do inglês) ou

conjuntos.

Para nosso estudo vamos unificar o nome de **vetor** para agrupamento de 1 dimensão e **matrizes** para agrupamentos de mais de uma dimensão.

Matrizes de Uma Dimensão - Vetor

A matriz de uma dimensão (Vetor) é a forma mais simples de usar tabelas de valores com apenas uma coluna e varias linhas de dados.

Linha	Coluna
1	
2	
3	
4	
:	

Matrizes de Uma Dimensão - Vetor

Estruturas indexadas homogêneas - Vetor

Exemplo:

índice:	0	1	2	3	4	5	6	7	8	9
nome:										

onde:

nome : identificação da variável

índice: posição em que está armazenado o dado

Essa estrutura de dados fica em uma única variável dimensionada com um determinado tamanho

A dimensão de uma matriz é formada por constantes inteiras e positivas.

Os nomes dados a uma variável composta (matriz) seguem as mesmas regras dos nomes dados a variáveis simples

Matrizes de Uma Dimensão - Vetor

Definição de dados indexados:

A forma geral para se definir uma variável agrupada é a mesma de um variável simples, seguida pela definição de elementos do grupo, por dimensão.

Em Matlab o primeiro valor ocupará a posição de índice igual a um (1)

Em C o primeiro valor ocupará a posição de índice igual a zero(0)

Em algoritmo, tanto *<valor-inicial>* como *<valor-final>* devem ser inteiros. Além disso, exige-se evidentemente que *<valor-final>* seja maior do que *<valor-inicial>*.

Exemplos:

vet: *vetor* [0..4] de real // reserva 5 posições de memória variável vet

qtd: *vetor* [0..9] de inteiro // vetor de 10 posições para qtd

Em C

float vet[5]; // reserva 5 posições de memória variável vet índices de 0 a 4

int qtd[10]; // vetor de 10 posições para qtd índices de 0 a 9

Matrizes de Uma Dimensão - Vetor

Exemplos:

v: **vetor** [0..9] de **real** // reserva 10 posições de memória variável V

Após preenchendo o vetor

variavel : V

índice	0	1	2	3	4	5	6	7	8	9
valor	58	4	2	123	-4	1	59	500	857	8

variavel : V

índice	0	1	2	3	4	5	6	7	8	9
elemento	1º	2º	3º	4º	5º	6º	7º	8º	9º	10º
variavel	var[0]	var[1]	var[2]	var[3]	var[4]	var[5]	var[6]	var[7]	var[8]	var[9]
valor	58	4	2	123	-4	1	59	500	857	8

Matrizes de Uma Dimensão - Vetor

Acesso a elementos: é individual para cada elemento e deverá ser feito mediante o uso de colchetes [] envolvendo uma constante, uma variável inteira ou por expressão cujo valor resulte também em um valor inteiro.

Exemplos:

```
algoritmo "semnome"
// Função :
// Autor :
// Data :
var
  p: vetor [0..2] de inteiro
  // p vetor de 3 posicoes p[0],p[1] e p[2]
  a,b,i:inteiro
inicio
  p[0]<-5
  a<-1
  b<-1
  p[a]<-8
  p[a+b]<-13
  para i de 0 ate 2 passo 1 faca
    escreval(p[i])
  fimpara
finalgoritmo
```

P	0	5		
	1	8		
	2	13		
a	1			5
b	1			8
i	0	1	2	13

Matrizes de Uma Dimensão - Vetor

Refazer um algoritmo para ler três notas de alunos e imprimir quantas notas tem valor abaixo da média utilizando vetores

```
7  Var
8      n1,n2,n3,media:real
9      nr:inteiro
10 Inicio
11     // Seção de Comandos
12     leia(n1,n2,n3)
13     media<-(n1+n2+n3)/3
14     nr<-0
15     se n1<media entao
16         nr<-nr+1
17     fimse
18     se n2<media entao
19         nr<-nr+1
20     fimse
21     se n3<media entao
22         nr<-nr+1
23     fimse
24     escreva(nr)
25 Fimalgoritmo
```

```
Var
notas: vetor [0..2] de real
media,soma:real
i,nr,tam:inteiro
Inicio
// Seção de Comandos
tam <-3
soma <-0
para i de 0 ate tam-1 passo 1 faca
    leia(notas[i])
    soma<-soma+notas[i]
fimpara
media <-soma/tam
nr<-0
para i de 0 ate tam-1 passo 1 faca
    se notas[i] < media entao
        nr<-nr+1
    fimse
fimpara
escreva(nr)
Fimalgoritmo
```

Matrizes de Uma Dimensão - Vetor

Refazer um algoritmo para ler três notas de alunos e imprimir quantas notas tem valor abaixo da média utilizando vetores

Var

```
notas: vetor [0..2] de real  
media, soma: real  
i, nr, tam: inteiro
```

Inicio

// Seção de Comandos

```
tam <- 3  
soma <- 0  
para i de 0 ate tam-1 passo 1 faca  
    leia(notas[i])  
    soma <- soma + notas[i]  
fimpara  
media <- soma / tam  
nr <- 0  
para i de 0 ate tam-1 passo 1 faca  
    se notas[i] < media entao  
        nr <- nr + 1  
    fimse  
fimpara  
escreva(nr)
```

Fimalgoritmo

```
1  #include <stdio.h>  
2  #include <stdlib.h>  
3  int main()  
4  {  
5      const int tam=3;  
6      float notas[tam];  
7      float media, soma;  
8      int i, nr;  
9  
10     soma=0;  
11     for (i=0; i<tam; i++) {  
12         printf("Digite a %i nota:", i+1);  
13         scanf("%f", &notas[i]);  
14         soma = soma + notas[i];  
15     }  
16     media=soma/tam;  
17     nr=0;  
18     for (i=0; i<tam; i++) {  
19         if (notas[i]<media) {  
20             nr=nr+1;  
21         }  
22     }  
23     printf("%i aluno(s) possui(em) .\n", nr);  
24     return 0;  
25 }
```


Matrizes de Uma Dimensão - Vetor

Exercícios:

Ler um vetor de 10 posições. Imprimir a soma dos valores do vetor.

Exemplo:

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  int main()
4  {
5      const int tam=3;
6      float notas[tam];
7      float media,soma;
8      int i,nr;
9
10     soma=0;
11     for (i=0;i<tam;i++){
12         printf("Digite a %i nota:",i+1);
13         scanf("%f",&notas[i]);
14         soma = soma+ notas[i];
15     }
16     media=soma/tam;
17     nr=0;
18     for (i=0;i<tam;i++){
19         if (notas[i]<media){
20             nr=nr+1;
21         }
22     }
23     printf("%i aluno(s) possui(em).\n",nr);
24     return 0;
25 }
```

Matrizes de Uma Dimensão - Vetor

Exercícios:

Ler um vetor de 10 posições. Imprimir a soma dos valores do vetor.

```
var  
  vet:vetor[0..9] de inteiro  
  i,soma:inteiro  
inicio  
  para i de 0 ate 9 passo 1 faca  
    leia(vet[i])  
  fimpara  
  soma<-0  
  para i de 0 ate 9 passo 1 faca  
    soma<-soma+vet[i]  
  fimpara  
  escreva("A soma do conjunto é ",soma)  
fimalgoritmo
```

Nome	Tipo	Valor
VET[0]	I	1
VET[1]	I	2
VET[2]	I	3
VET[3]	I	3
VET[4]	I	2
VET[5]	I	1
VET[6]	I	1
VET[7]	I	2
VET[8]	I	3
VET[9]	I	1
I	I	10
SOMA	I	19

```
1  
2  
3  
3  
2  
1  
1  
2  
3  
1  
A soma do conjunto é 19
```

Matrizes de Uma Dimensão - Vetor

Exercícios:

Ler um vetor de 10 posições. Imprimir a soma dos valores do vetor.

```
var  
  vet:vetor[0..9] de inteiro  
  i,soma:inteiro  
inicio  
  para i de 0 ate 9 passo 1 faca  
    leia(vet[i])  
  fimpara  
  soma<-0  
  para i de 0 ate 9 passo 1 faca  
    soma<-soma+vet[i]  
  fimpara  
  escreva("A soma do conjunto é ",soma)  
fimalgoritmo
```

```
1  #include <stdio.h>  
2  #include <stdlib.h>  
3  #include <locale.h>  
4  
5  int main()  
6  {  
7      setlocale(LC_ALL,"portuguese");  
8      int vet[10],i,soma;  
9      for(i=0; i<10; i=i+1)  
10     {  
11         printf("Digite o %d valor:",i);  
12         scanf("%d",&vet[i]);  
13     }  
14     soma=0;  
15     for(i=0; i<10; i=i+1)  
16     {  
17         soma=soma+vet[i];  
18     }  
19     printf("A soma do conjunto é %d\n",soma);  
20     return 0;  
21 }
```

```
Digite o 0 valor:1  
Digite o 1 valor:2  
Digite o 2 valor:3  
Digite o 3 valor:3  
Digite o 4 valor:2  
Digite o 5 valor:1  
Digite o 6 valor:1  
Digite o 7 valor:2  
Digite o 8 valor:3  
Digite o 9 valor:1  
A soma do conjunto é 19
```

Matrizes de Uma Dimensão - Vetor

Exercícios:

Ler um vetor de 10 posições. Após a leitura pedir ao usuário se quer 1 – soma dos números pares do vetor, ou 2 soma dos números ímpares do vetor, e imprimir a resposta

Queremos saber se o número contido no vetor é par ou ímpar

Matrizes de Uma Dimensão - Vetor

Exercícios:

Ler um vetor de 10 posições. Após a leitura pedir ao usuário se quer 1 – soma dos números pares do vetor, ou 2 soma dos números ímpares do vetor, e imprimir a resposta

```
var
vet:vetor[0..9] de inteiro
i,op,soma:inteiro
inicio
para i de 0 ate 9 passo 1 faca
    leia(vet[i])
fimpara
escreva("1 - para somar numeros impares, 2 para somar numeros pares do veto")
leia(op)
soma<-0
para i de 0 ate 9 passo 1 faca
    se (op=1) e (vet[i] % 2 =1) entao
        soma<-soma+vet[i]
    senao
        se (op=2) e (vet[i] % 2 =0) entao
            soma<-soma+vet[i]
        fimse
    fimse
fimpara
se (op=1) entao
    escreva("A soma dos numeros impares do conjunto é ",soma)
senao
    escreva("A soma dos numeros pares do conjunto é ",soma)
fimse
finalgoritmo
```

```
1
2
3
3
2
1
1
2
3
3
1 - para somar numeros impares, 2 para somar numeros pares do veto1
A soma dos numeros impares do conjunto é 15
```

```
1 - para somar numeros impares, 2 para somar numeros pares do veto2
A soma dos numeros pares do conjunto é 6
*** Fim da execução.
*** Feche esta janela para retornar ao Visualg.
```

Exercícios:

Ler um vetor de 10 posições. Após a leitura pedir ao usuário se quer 1 – soma dos números pares do vetor, ou 2 soma dos números ímpares do vetor, e imprimir a resposta

```
var
vet:vetor[0..9] de inteiro
i,op,soma:inteiro
inicio
  para i de 0 ate 9 passo 1 faca
    leia(vet[i])
  fimpara
  escreva("1 - para somar numeros impares, 2 para somar numeros pares")
  leia(op)
  soma<-0
  para i de 0 ate 9 passo 1 faca
    se (op=1) e (vet[i] % 2 =1) entao
      soma<-soma+vet[i]
    senao
      se (op=2) e (vet[i] % 2 =0) entao
        soma<-soma+vet[i]
      fimse
    fimse
  fimpara
  se (op=1) entao
    escreva("A soma dos numeros impares do conjunto é ",soma)
  senao
    escreva("A soma dos numeros pares do conjunto é ",soma)
  fimse
fimalgoritmo
```

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <locale.h>
4  int main()
5  {
6      setlocale(LC_ALL,"portuguese");
7      int vet[10],i,op,soma;
8      for(i=0; i<10; i=i+1)
9      {
10         printf("Digite o %d valor:",i);
11         scanf("%d",&vet[i]);
12     }
13     printf("Escolha 1 para somar os numeros impares do vetor");
14     printf(" 2 para somar os numeros pares do vetor:");
15     scanf("%d",&op);
16     soma=0;
17     for(i=0; i<10; i=i+1)
18     {
19         if ((op==1) && (vet[i]%2==1))
20         {
21             soma=soma+vet[i];
22         }
23         else
24         {
25             if ((op==2) && (vet[i]%2==0))
26             {
27                 soma=soma+vet[i];
28             }
29         }
30     }
31     if (op==1)
32     {
33         printf("A soma dos números impares do conjunto é %d\n",soma);
34     }
35     else
36     {
37         printf("A soma dos números pares do conjunto é %d\n",soma);
38     }
39     return 0;
40 }
```

Matrizes de Uma Dimensão - Vetor

Exercícios:

Ler um vetor de 10 posições. Após a leitura pedir ao usuário se quer 1 – soma dos valores dos índices pares do vetor, ou 2 soma dos valores dos índices ímpares do vetor, e imprimir a resposta

Queremos saber se o índice do vetor e não o número contido no vetor é par ou ímpar

Quando se faz menção ao índice, indica-se a variável que controla o contador de índice. Quando se faz menção ao elemento, indica-se $A[i]$, pois desta forma pega-se o valor armazenado e não a sua posição de endereço.

Exercícios:

Ler um vetor de 10 posições. Após a leitura pedir ao usuário se quer 1 – soma dos valores dos índices pares do vetor, ou 2 soma dos valores dos índices ímpares do vetor, e imprimir a resposta

```
var
  vet:vetor[0..9] de inteiro
  i,op,soma:inteiro
inicio
  para i de 0 ate 9 passo 1 faca
    leia(vet[i])
  fimpara
  escreva("1 - para somar valores dos índices ímpares,")
  escreva(" 2 para somar valores dos índices pares do vetor")
  leia(op)
  soma<-0
  para i de 0 ate 9 passo 1 faca
    se (op=1) e (i % 2 =1) entao
      soma<-soma+vet[i]
    senao
      se (op=2) e (i % 2 =0) entao
        soma<-soma+vet[i]
      fimse
    fimse
  fimpara
  se (op=1) entao
    escreva("A soma dos numeros dos índices ímpares do conjunto é ",soma)
  senao
    escreva("A soma dos numeros dos índices pares do conjunto é ",soma)
  fimse
finalgoritmo
```

```
1
2
3
3
2
1
1
2
3
3
1 - para somar valores dos índices ímpares, 2 para somar valores dos índices pares
A soma dos numeros dos índices ímpares do conjunto é 11

1 - para somar valores dos índices ímpares, 2 para somar valores dos índices pares
A soma dos numeros dos índices pares do conjunto é 10
```


Exercícios:

Ler um vetor de 10 posições. Após a leitura pedir ao usuário se quer 1 – soma dos valores dos índices pares do vetor, ou 2 soma dos valores dos índices ímpares do vetor e imprimir a resposta

```
var
  vet:vetor[0..9] de inteiro
  i,op,soma:inteiro
inicio
  para i de 0 ate 9 passo 1 faca
    leia(vet[i])
  fimpara
  escreva("1 - para somar valores dos índices ímpares,")
  escreva(" 2 para somar valores dos índices pares do vetor")
  leia(op)
  soma<-0
  para i de 0 ate 9 passo 1 faca
    se (op=1) e (i % 2 =1) entao
      soma<-soma+vet[i]
    senao
      se (op=2) e (i % 2 =0) entao
        soma<-soma+vet[i]
      fimse
    fimse
  fimpara
  se (op=1) entao
    escreva("A soma dos numeros dos índices ímpares do conjunto é ")
  senao
    escreva("A soma dos numeros dos índices pares do conjunto é ")
  fimse
finalgoritmo
```

```
2  #include <stdlib.h>
3  #include <locale.h>
4  int main()
5  {
6      setlocale(LC_ALL,"portuguese");
7      int vet[10],i,op,soma;
8      for(i=0; i<10; i=i+1)
9      {
10         printf("Digite o %d valor:",i);
11         scanf("%d",&vet[i]);
12     }
13     printf("Escolha 1 para somar os numeros dos índices ímpares do vetor");
14     printf(" 2 para somar os numeros dos índices pares do vetor:");
15     scanf("%d",&op);
16     soma=0;
17     for(i=0; i<10; i=i+1)
18     {
19         if ((op==1) && (i%2==1))
20         {
21             soma=soma+vet[i];
22         }
23         else
24         {
25             if ((op==2) && (i%2==0))
26             {
27                 soma=soma+vet[i];
28             }
29         }
30     }
31     if (op==1)
32     {
33         printf("A soma dos números dos índices ímpares do conjunto é %d\n",soma);
34     }
35     else
36     {
37         printf("A soma dos números dos índices pares do conjunto é %d\n",soma);
38     }
39     return 0;
40 }
```

Matrizes de Uma Dimensão - Vetor

- Construa um procedimento para ler um vetor real de tamanho N – passe o vetor por referência, e o tamanho N por valor.
- Construa um procedimento para imprimir um vetor real de tamanho N – passe o vetor e o tamanho N por valor.

Escrever um programa que leia dois vetores (denominados A e B) , utilizando o procedimento criado, com 5 elementos reais. Construir um vetor C, sendo cada elemento de C a subtração de um elemento correspondente do vetor A com um elemento correspondente ao vetor B, ou seja, a operação de processamento deve estar baseada na operação $C[I] \leftarrow A[I] - B[I]$. Ao final, imprimir utilizando o procedimento criado, os elementos do vetor C.

Matrizes de Uma Dimensão - Vetor

- Construa um procedimento para ler um vetor real de tamanho N – passe o vetor por referência, e o tamanho N por valor.

```
// segue as convenções
procedimento leVetor (var v:vetor[0..N-1] de real, N:inteiro)
var
  i:inteiro
inicio
  para i de 0 ate N-1 passo 1 faca
    leia (v[i])
  fimpara
fimprocedimento
```

Passagem por referência

```
17 //
18 void leVetor (float *V, int tam)
19 {
20     int i;
21     for (i=0;i<tam;i=i+1)
22     {
23         printf("Digite o %d termo do vetor:",i);
24         scanf("%f",&V[i]);
25     }
26 }
```

Matrizes de Uma Dimensão - Vetor


- Construa um procedimento para imprimir um vetor real de tamanho N – passe o vetor e o tamanho N por valor.

```
procedimento imprimeVetor (v:vetor[0..N-1] de real, N:inteiro)  
var  
    i:inteiro  
inicio  
    para i de 0 ate N-1 passo 1 faca  
        escreva (v[i])  
    fimpara  
fimprocedimento
```

28
29
30
31
32
33
34
35

```
void imprimeVetor(float *V, int tam)  
{  
    int i;  
    for (i=0;i<tam;i=i+1)  
    {  
        printf("V[%d]=%5.2f\n",i,V[i]);  
    }  
}
```

Em C os vetores são
sempre por
referência



Matrizes de Uma Dimensão - Vetor

Escrever um programa que leia dois vetores (denominados A e B) , utilizando o procedimento criado, com 5 elementos reais. Construir um vetor C, sendo cada elemento de C a subtração de um elemento correspondente do vetor A com um elemento correspondente ao vetor B, ou seja, a operação de processamento deve estar baseada na operação $C[I] \leftarrow A[I] - B[I]$. Ao final, imprimir utilizando o procedimento criado, os elementos do vetor C.

```
var
  A,B,C:vetor[0..4] de inteiro
  i,soma:inteiro
inicio
  leVetor(A,5)
  leVetor(B,5)
  para i de 0 ate 4 passo 1 faca
    C[i]<-A[i]-B[i]
  fimpara
  imprimeVetor(C,5)
fimalgoritmo
```

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  void leVetor (float *, int);
4  void imprimeVetor(float *, int);
5  int main()
6  {
7      float A[5],B[5],C[5];
8      int i;
9      leVetor(A,5);
10     leVetor(B,5);
11     for (i=0;i<5;i=i+1)
12     {
13         C[i]=A[i]-B[i];
14     }
15     imprimeVetor(C,5);
16     return 0;
17 }
```

Exercícios - arranjos

`srand(time(NULL));` // Inicialização, deve ser chamado uma única vez.

`int r = rand();` // retorna um número randômico inteiro entre 0 e `RAND_MAX`

Para os exercícios abaixo crie uma função utilizando `rand` – que fornecido os valores inicial e final devolva um número aleatório dentro deste intervalo.

1 - Preencher um vetor de 15 posições com valores aleatórios de 15 a 50. Construir um vetor B do mesmo tipo, em que cada elemento de B deva ser o resultado do fatorial correspondente de cada elemento da matriz A. Apresentar A e B.

2 - Preencher dois vetores A e B de 15 posições cada com valores aleatórios de `vi` a `vf` (lidos do teclado). Construir um vetor C, sendo este o resultado da união dos elementos de A e B – sem repetição. Apresentar C.

3 - Preencher um vetor A de 20 elementos com valores aleatórios de `vi` a `vf` (lidos do teclado), em seguida crie um procedimento que inverta os elementos armazenados. Ou seja, o primeiro elemento de A passará a ser o último, o segundo elemento passará a ser o penúltimo e assim por diante. Apresentar A.

4 - Preencher um vetor A de x elementos (x deverá ser lido do teclado) com valores aleatórios de `vi` a `vf` (lidos do teclado). Crie um vetor `ParImpar` de 2 posições e armazene no índice 0 quantos elementos de A são par e no índice 1 quantos elementos de A são ímpar. Apresentar o vetor `ParImpar`. Obs.: não utilize nenhum comando condicional(`if,switch`).

5 - Preencher um vetor A de 10 elementos com valores aleatórios de `vi` a `vf` (lidos do teclado). Ordene e imprima o vetor A.



PUC Minas
Virtual