

# CK0211 - Fundamentos de Programação: Vetores

Emanuele Santos

Bibliografia: Ascencio, Cap. 6

# Objetivos

- Aprofundar o conceito de Vetor em Algoritmos através de exemplos resolvidos e exercícios

# Relembrando...

- Declaração de vetores

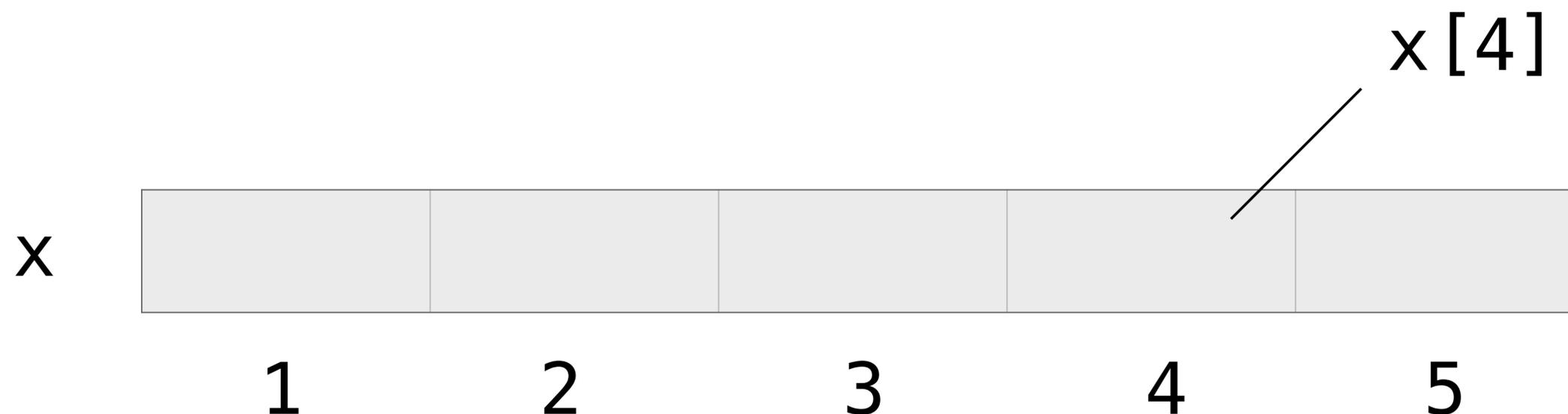
```
DECLARE nome[tamanho] TIPO
```

- onde:
  - nome é o nome da variável do tipo vetor;
  - tamanho é a quantidade de variáveis que vão compor o vetor;
  - tipo é o tipo básico de cada elemento do vetor

# Exemplo de vetor

```
DECLARE x[5] NUMÉRICO
```

- $x$  é um vetor com 5 posições
- para acessar um valor em uma determinada posição  $i$ , usamos  $x[i]$



# Exemplo resolvido 1

- Faça um programa que preencha um vetor com cinco números inteiros, determine e mostre os números pares e suas respectivas posições
- Estratégia:
  - 1) Usar um laço para preencher o vetor com os números digitados pelo usuário
  - 2) Usar outro laço para percorrer o vetor e para cada elemento do vetor verificar se o número é par e se sim, mostrar o elemento e a posição dele no vetor

```
ALGORITMO  
DECLARE num[5], i, j, cont NUMÉRICO  
PARA i ← 1 ATÉ 5 FAÇA  
INÍCIO  
    LEIA num[i]  
FIM  
PARA i ← 1 ATÉ 5 FAÇA  
INÍCIO  
    SE num[i] % 2 = 0 ENTÃO  
        INÍCIO  
            ESCREVA num[i]  
            ESCREVA i  
        FIM  
FIM  
FIM  
FIM_ALGORITMO.
```

# Exemplo resolvido 2

*Ascencio, Exercício 1, pág. 159*

- Faça um programa que preencha um vetor com nove números inteiros, calcule e mostre os números primos e suas respectivas posições.
- Estratégia:
  - 1) Usar um laço para preencher o vetor com os números digitados pelo usuário
  - 2) Usar outro laço para percorrer o vetor e para cada elemento do vetor verificar se o número é primo e se sim, mostrar o elemento e a posição dele no vetor

# ALGORITMO

DECLARE num[9], i, j, cont NUMÉRICO

PARA i ← 1 ATÉ 9 FAÇA

INÍCIO

    LEIA num[i]

FIM

PARA i ← 1 ATÉ 9 FAÇA

INÍCIO

    cont ← 0

    PARA j ← 1 ATÉ num[i] FAÇA

    INÍCIO

        SE num[i] % j = 0 ENTÃO cont ← cont + 1

    FIM

    SE cont = 2 ENTÃO

    INÍCIO

        ESCREVA num[i]

        ESCREVA i

    FIM

FIM

FIM\_ALGORITMO.

# Exercício 1

- Faça um programa que preencha um vetor com 10 elementos numéricos inteiros. Calcule e mostre:
  - todos os números pares;
  - a quantidade de números pares;
  - todos os números ímpares;
  - a quantidade de números ímpares.

# Exemplo resolvido 3

*Ascencio, Exercício 2, pág. 159*

- Uma pequena loja de artesanato possui apenas um vendedor e comercializa dez tipos de objetos. O vendedor recebe, mensalmente, salário de R\$545,00, acrescido de 5% do valor total de suas vendas. O valor unitário dos objetos deve ser informado e armazenado em um vetor; a quantidade vendida de cada peça deve ficar em outro vetor, mas na mesma posição. Crie um programa que receba os preços e as quantidades vendidas, armazenando-os em seus respectivos vetores (ambos com tamanho dez). Depois determine e mostre:
  - um relatório contendo: a quantidade vendida, valor unitário e valor total de cada objeto. Ao final, deverão ser mostrados o valor geral das vendas e o valor da comissão que será paga ao vendedor; e
  - o valor do objeto mais vendido e sua posição no vetor (não se preocupe com empates).

## ALGORITMO

**DECLARE** preco[10], qtd[10] **NUMÉRICO**

        i, tot\_geral, tot\_vend, comissao, maior, imaior **NUMÉRICO**

tot\_geral ← 0

**PARA** i ← 1 **ATÉ** 10 **FAÇA**

**INÍCIO**

**LEIA** preco[i]

**LEIA** qtd[i]

**FIM**

**PARA** i ← 1 **ATÉ** 10 **FAÇA**

**INÍCIO**

    tot\_vend ← qtd[i] \* preco[i]

**ESCREVA** qtd[i], preco[i], tot\_vend

    tot\_geral ← tot\_geral + tot\_vend

**FIM**

comissao ← tot\_geral \* 5 / 100

**ESCREVA** tot\_geral, comissao

...

```
...  
maior ← qtd[1]  
imaior ← 1  
PARA i ← 2 ATÉ 10 FAÇA  
INÍCIO  
  SE qtd[i] > maior ENTÃO  
    INÍCIO  
      maior ← qtd[i]  
      imaior ← i  
    FIM  
FIM  
ESCREVA preco[imaior], maior  
FIM_ALGORITMO.
```

# Exercício 2

- Faça um programa para controlar o estoque de mercadorias de uma empresa. Inicialmente, o programa deverá preencher dois vetores com dez posições cada, onde o primeiro corresponde ao código do produto e o segundo, ao total desse produto em estoque. Logo após, o programa deverá ler um conjunto indeterminado de dados contendo o código de um cliente e o código do produto que ele deseja comprar, juntamente com a quantidade. Código do cliente igual a zero indica fim do programa. O programa deverá verificar:
  - se o código do produto solicitado existe. Se existir, tentar atender ao pedido; caso contrário, exibir mensagem **Código inexistente**;
  - cada pedido feito por um cliente só pode ser atendido integralmente. Caso isso não seja possível, escrever a mensagem **Não temos estoque suficiente dessa mercadoria**. Se puder atendê-lo, escrever a mensagem **Pedido atendido. Obrigado e volte sempre**;
  - efetuar a atualização do estoque somente se o pedido for atendido integralmente;
  - no final do programa, escrever os códigos dos produtos com seus respectivos estoques já atualizados.

# Exemplo resolvido 4

*Ascencio, Exercício 4, pág. 161*

- Faça um programa que preencha um vetor com oito números inteiros, calcule e mostre dois vetores resultantes. O primeiro vetor resultante deve conter os números positivos e o zero, e o segundo, os números negativos. Cada vetor resultante vai ter, no máximo, oito posições, que não poderão ser completamente utilizadas.
- Estratégia:
  - 1) Usar um laço para preencher o vetor com os números digitados pelo usuário e no mesmo laço já copia o número para o vetor resultante correspondente; atualizar o número de posições utilizadas no vetor resultante
  - 2) Depois percorrer cada vetor resultante mostrando os valores.

## ALGORITMO

**DECLARE** num[8], pos[8], neg[8] NUMÉRICO

    i, cont\_p, cont\_n NUMÉRICO

cont\_p ← 1

cont\_n ← 1

**PARA** i ← 1 **ATÉ** 8 **FAÇA**

**INÍCIO**

**LEIA** num[i]

**SE** num[i] ≥ 0 **ENTÃO**

**INÍCIO**

            pos[cont\_p] ← num[i]

            cont\_p ← cont\_p + 1

**FIM**

**SENÃO**

**INÍCIO**

            neg[cont\_n] ← num[i]

            cont\_n ← cont\_n + 1

**FIM**

**FIM**

...

```
...  
SE cont_p = 1 ENTÃO ESCREVA "Vetor de positivos vazio"  
SENÃO  
INÍCIO  
  PARA i ← 1 ATÉ cont_p-1 FAÇA  
  INÍCIO  
    ESCREVA pos[i]  
  FIM  
FIM  
SE cont_n = 1 ENTÃO ESCREVA "Vetor de negativos vazio"  
SENÃO  
INÍCIO  
  PARA i ← 1 ATÉ cont_n-1 FAÇA  
  INÍCIO  
    ESCREVA neg[i]  
  FIM  
FIM  
FIM_ALGORITMO.
```

# Exemplo resolvido 5

- Faça um programa que preencha dois vetores,  $X$  e  $Y$ , com dez números inteiros cada. Calcule e mostre um vetor resultante contendo a Interseção entre  $X$  e  $Y$  (apenas os elementos que aparecem nos dois vetores, sem repetições).
- Estratégia:
  - 1) Usar um laço para preencher os dois vetores  $X$  e  $Y$
  - 2) Depois percorrer o vetor  $X$ , verificando se cada elemento de  $X$  também está em  $Y$ . Se estiver, preencher o vetor de interseção, se o elemento já não estiver lá.

```
ALGORITMO  
DECLARE X[10], Y[10], IT[10] NUMÉRICO  
         cont_it, i, j, k NUMÉRICO  
PARA i ← 1 ATÉ 10 FAÇA  
INÍCIO  
    LEIA X[i]  
    LEIA Y[i]  
FIM  
...
```

```
cont_it ← 1
PARA i ← 1 ATÉ 10 FAÇA
INÍCIO
  j ← 1
  ENQUANTO j ≤ 10 E X[i] ≠ Y[j] FAÇA j ← j + 1
  SE j ≤ 10 ENTÃO
    INÍCIO
      k ← 1
      ENQUANTO k < cont_it E X[i] ≠ IT[k] FAÇA k ← k + 1
      SE k ≥ cont_it ENTÃO
        INÍCIO
          IT[cont_it] ← X[i]
          cont_it ← cont_it + 1
        FIM
      FIM
    FIM
  FIM
FIM
PARA i ← 1 ATÉ cont_it-1 FAÇA
INÍCIO
  ESCREVA IT[i]
FIM
FIM_ALGORITMO.
```

# Exercício 3

- A coordenação da Ciência da Computação deseja saber se existem alunos cursando, simultaneamente, as disciplinas de Fundamentos de Programação (FUP) e Cálculo II. Coloque os números das matrículas dos alunos que cursam FUP em um vetor, 20 alunos. Coloque os números das matrículas dos alunos que cursam Cálculo II em outro vetor, 40 alunos. Mostre o número das matrículas que aparecem nos dois vetores.

# Tarefa SIGAA

- Faça os programas equivalentes em Python para cada um dos exemplos e exercícios da aula de hoje. Crie uma pasta chamada **vetores** dentro da pasta FUP e crie os programas usando nomes de arquivo abaixo na pasta **vetores**. Ao final, crie um arquivo compactado **.zip** da pasta vetores e envie pelo SIGAA.
- exemplo1.py, exemplo2.py, exemplo3.py, exemplo4.py, exemplo5.py, exerc1.py, exerc2.py e exerc3.py.
- Prazo de entrega: até 16/10/2016, 23:59h