

Programação Aplicada à Engenharia

Professora Priscila Marques Kai
Aula 5

Estrutura Condicional

Estrutura SE-ENTÃO-SENÃO:

SE <expressão booleana> ENTÃO

<instruções a serem executadas caso a expressão booleana resulte em VERDADEIRO>

SENÃO

<instruções a serem executadas caso a expressão booleana resulte em FALSO>

FIM-SE

Indentação

A **indentação** possui a função de deixar o código mais fácil de entender, criando uma hierarquia de comandos que serão executados em um algoritmo. Para isso, espaços podem ser utilizados, construindo alinhamentos no código dentro de outro.

Assim, a indentação indica o início e o final de blocos de código.

Algoritmos e suas representações

ALGORITMO Calcula_raizes

var

A: INTEIRO

B: INTEIRO

C: INTEIRO

INICIO

LER A, B, C

SE $A = 0$ ENTÃO

ESCREVA ("Não é equação de 2º grau")

SENÃO

$$D \leftarrow (B^{**2}) - 4*A*C$$

SE $D < 0$ ENTÃO

```
ESCREVA("Não existem raízes reais!")
```

SENÃO

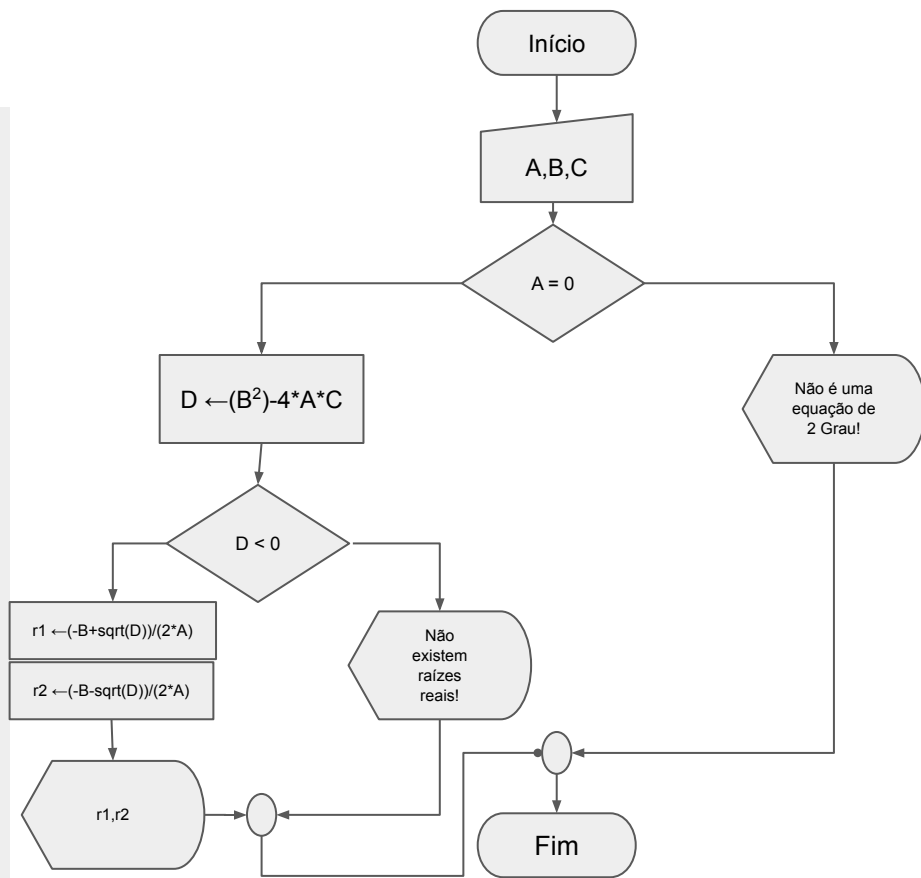
$$r1 = (-B + \text{sqrt}(D)) / (2*A)$$
$$r_2 = (-B - \sqrt{D}) / (2*A)$$

```
ESCREVA("r1: ", r1, "r2: ", r2)
```

FIMSE

FIMSE

FIM



Algoritmos e suas representações

```
from math import sqrt
```

```
A = 1
```

```
B = 2
```

```
C = 3
```

```
if A == 0:
```

```
    print("Não é equação de 2 grau")
```

```
else:
```

```
    D = (B**2) - 4*A*C
```

```
    if D < 0:
```

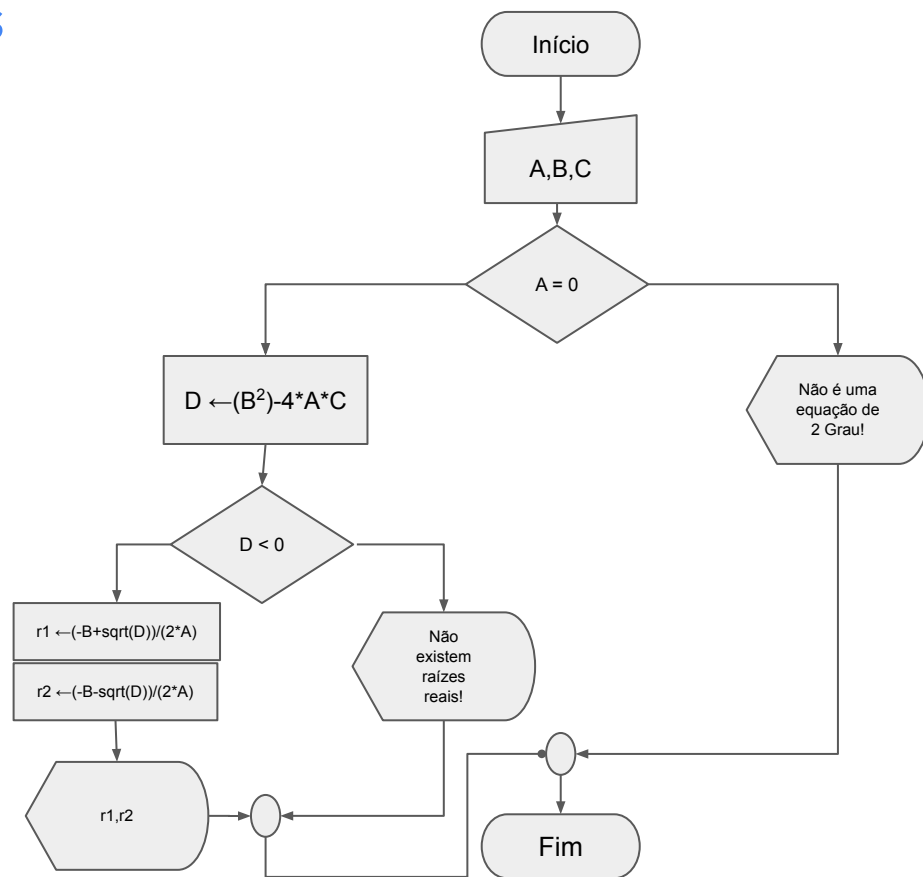
```
        print("Não existem raízes reais!")
```

```
    else:
```

```
        r1 = (-B + sqrt(D)) / (2*A)
```

```
        r2 = (-B - sqrt(D)) / (2*A)
```

```
        print("r1: ", r1, "r2: ", r2)
```



Exercícios

- 1 - Faça um algoritmo que receba como entrada as **quatro notas** de um aluno do semestre. Após, calcule a média das notas e imprima na tela se o aluno foi aprovado (média ≥ 6), ou reprovado (média < 6).
- 2 - Crie um fluxograma do algoritmo do exercício anterior.
- 3 - Crie um sistema de classificação de números inteiros com base em seus valores. O sistema deve receber como entrada 3 números e em seguida, classificá-los do maior para o menor número. No final o algoritmo deve imprimir os números em ordem decrescente.
- 4 - Crie um fluxograma do algoritmo do exercício anterior.
- 5 - Você é um engenheiro de produção em uma fábrica que produz um determinado produto. Sua tarefa é criar um algoritmo para controlar o estoque de matéria-prima na fábrica. O programa deve permitir o registro de entradas e saídas de materiais no estoque e manter o controle dos níveis de estoque disponíveis. As regras são as seguintes:
 - Se a quantidade de material no estoque for menor que 100 unidades, é necessário solicitar uma nova compra de matéria-prima.
 - Se a quantidade de material no estoque for maior ou igual a 100 unidades, o estoque está adequado.
- 6 - Crie um fluxograma do algoritmo do exercício anterior.

Exercícios

1 - Faça um algoritmo que receba como entrada as três notas de um aluno do semestre. Após, calcule a média das notas e imprima na tela se o aluno foi aprovado (média ≥ 6), ou reprovado (média < 6).

```
ALGORITMO "Calcula_Média"
var
    Nota1 : REAL
    Nota2 : REAL
    Nota3 : REAL
    Nota4 : REAL
INÍCIO
    LER Nota1, Nota2, Nota3, Nota4
    Media = (Nota1 + Nota2 + Nota3 + Nota4) / 4
    SE Media  $\geq$  6 ENTÃO
        ESCREVER("Aluno Aprovado!")
    SENÃO
        ESCREVER("Aluno Reprovado!")
    FIMSE
FIM
```

Exercícios

3 - Crie um sistema de ordenação de números inteiros com base em seus valores. O sistema deve receber como entrada 3 números e em seguida, ordenando-os do maior para o menor número. No final o algoritmo deve imprimir os números em ordem decrescente.

Algoritmo Ordena_numeros

var

 numero1, numero2, numero3, aux : INTEIRO

INICIO

 aux \leftarrow 0

 // Lê os números

 LER numero1, numero2, numero3

 // Inicia as comparações

 SE (numero2 > numero1) ENTÃO

 aux \leftarrow numero1

Exercícios

5 - Você é um engenheiro de produção em uma fábrica que produz um determinado produto. Sua tarefa é criar um algoritmo para controlar o estoque de uma matéria-prima na fábrica. O programa deve permitir o registro de entradas e saídas do material no estoque e manter o controle dos níveis de estoque disponíveis. As regras são as seguintes:

- Se a quantidade do material no estoque a ser removido for maior do que a quantidade atual no estoque, é necessário solicitar uma nova compra de matéria-prima.

Algoritmo Controle_Estoque

var

 entrada: INTEIRO

 saida: INTEIRO

 estoque: INTEIRO

 opcao: INTEIRO

INICIO

 estoque ← 100 // Inicializa o estoque com um valor

 ESCREVER("Estoque Inicial:", estoque)

 ESCREVER("Que operação deseja fazer? ")

 ESCREVER("Para inserir mais itens no estoque, digite 1: ")

 ESCREVER("Para remover itens no estoque, digite 2: ")

 ESCREVER("Para listar a quantidade no estoque, digite 3: ")

 LER opcao

 SE opcao = 1 ENTÃO

 ESCREVER("Informe a quantidade de itens a serem adicionados: ")

 LER entrada // Registrar uma entrada

 estoque ← estoque + entrada

 ESCREVER("Entrada de", entrada, "unidades. Estoque atual:", estoque)

 SENÃO

 SE opcao = 2 ENTÃO

 LER saida // Lê a quantidade de itens a serem retirados

 SE (estoque - saida >= 0) ENTÃO

 estoque ← estoque - saida

 ESCREVER("Saída de", saida, "unidades. Estoque atual:", estoque)

 SENÃO

 ESCREVER("Erro: Saída excede o estoque disponível.")

 FIMSE

 SENÃO

 SE opcao = 3 ENTÃO

 ESCREVER("Estoque atual:", estoque)

 SENÃO

 ESCREVER("Opção inexistente")

 FIMSE

 FIMSE

 FIMSE

FIM