

OBJETIVOS DO CAPÍTULO

- Conceitos de: decisão, condição, operadores relacionais e lógicos, e opção
- Comandos do FORTRAN: IF – THEN – ELSE – END IF, SELECT CASE – CASE – END SELECT

7.1 programa7a.f90

Para inicializar as atividades deste capítulo, deve-se executar:

- 1) Para acessar o programa Fortran, no Windows: **Start, Programs, Fortran PowerStation 4.0, Microsoft Developer Studio**
- 2) No Fortran, seguindo o [procedimento-padrão](#), **criar um projeto** com o nome **programa7**
- 3) No Fortran, seguindo o [procedimento-padrão](#), **criar e inserir** no projeto o programa-fonte **programa7a.f90**
- 4) Dentro do espaço de edição do Fortran, na subjanela maior, **copiar** exatamente o texto em vermelho mostrado na **Tabela 7.1**.
- 5) Comentários sobre o programa:
 - a) Nesta e na próxima seção são vistas três formas de uso do comando IF (SE). A primeira é mostrada na Tabela 7.2 e usada em diversas linhas do programa7a.f90. O comando IF é usado para tomar decisões com base em condições. Se as condições são satisfeitas, executa-se o comando que está na mesma linha do IF, caso contrário, não.
 - b) Por exemplo, na linha **IF (A < B) WRITE(*,*) "A eh MENOR do que B"** do programa, se o valor de A for menor do que o valor de B, é executado o comando que está após o parênteses, no caso, escrever um comentário. Aquilo que está dentro de parênteses é chamado de condição, que pode ser simples ou composta. Para definir as condições, são usados os chamados operadores relacionais (Tabela 7.3) e lógicos (Tabela 7.4).
 - c) Na linha **IF ((A < 0) .AND. (B < 0)) WRITE(*,*) "A E B sao numeros NEGATIVOS"** do programa, tem-se um exemplo de condição composta. Cada condição simples é unida pelos operadores lógicos .AND. e .OR., que em português significam E e OU.
 - d) Nas condições, além de variáveis, podem ser usados diretamente valores ou expressões envolvendo valores e variáveis.
 - e) Nas demais linhas do programa, são testados todos os operadores relacionais e lógicos.
- 6) Executar **Build, Compile** para compilar o programa.

7) Gerar o programa-executável fazendo **Build, Build**.

Tabela 7.1 Programa7a.f90.

```
REAL A, B

WRITE(*,*) "Entre com os valores de A e B"
READ(*,*) A, B

IF ( A < B ) WRITE(*,*) "A eh MENOR do que B"

IF ( A <= B ) WRITE(*,*) "A eh MENOR OU IGUAL a B"

IF ( A > B ) WRITE(*,*) "A eh MAIOR do que B"

IF ( A >= B ) WRITE(*,*) "A eh MAIOR OU IGUAL a B"

IF ( A == B ) WRITE(*,*) "A eh IGUAL a B"

IF ( A /= B ) WRITE(*,*) "A eh DIFERENTE de B"

IF ( ( A < 0 ) .AND. ( B < 0 ) ) WRITE(*,*) "A E B sao numeros NEGATIVOS"

IF ( ( A > 0 ) .AND. ( B > 0 ) ) WRITE(*,*) "A E B sao numeros POSITIVOS"

IF ( ( A > 0 ) .OR. ( B > 0 ) ) WRITE(*,*) "A OU B eh um numero POSITIVO"

END
```

Tabela 7.2 Sintaxe 1 do comando IF.

```
IF ( condições ) COMANDO
```

onde "condições" representa uma ou várias condições que devem ser satisfeitas para que COMANDO seja executado.

8) Ao se executar o programa, através de **Build, Execute**, surge uma janela, mostrada na Figura 7.1, dentro da qual tem-se:

a) Na primeira linha, o comentário Entre com os valores de A e B, resultado do comando

```
WRITE(*,*) "Entre com os valores de A e B" do programa.
```

- b) Na segunda linha, o programa pára e fica aguardando que sejam fornecidos os valores das variáveis A e B, resultado do comando `READ(*,*) A, B` do programa. Para que o programa continue sua execução é necessário **digitar 1.0 2.0**, por exemplo, e, em seguida, **clique na tecla enter**.
- c) Em seguida são apresentados os comentários correspondentes aos comandos IF cujas condições foram satisfeitas para os valores de A e B.

Tabela 7.3 Operadores relacionais do FORTRAN.

Operador	Significado
<	menor
<=	menor ou igual
>	maior
>=	maior ou igual
==	igual
/=	diferente

Tabela 7.4 Operadores lógicos do FORTRAN.

Operador	Significado
.AND.	e
.OR.	ou

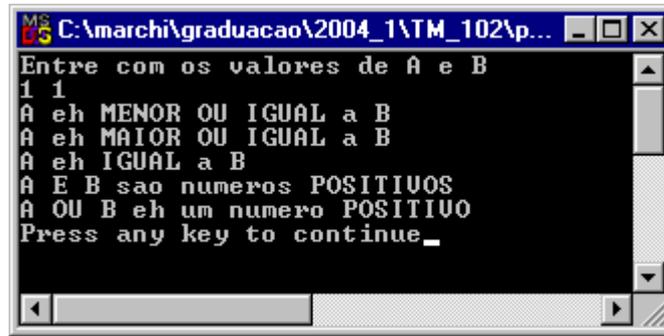
```

MS-DOS C:\marchi\graduacao\2004_1\TM_102\p...
Entre com os valores de A e B
1.0 2.0
A eh MENOR do que B
A eh MENOR OU IGUAL a B
A eh DIFERENTE de B
A E B sao numeros POSITIVOS
A OU B eh um numero POSITIVO
Press any key to continue_

```

Figura 7.1 Resultado do programa7a.f90 para A = 1.0 e B = 2.0.

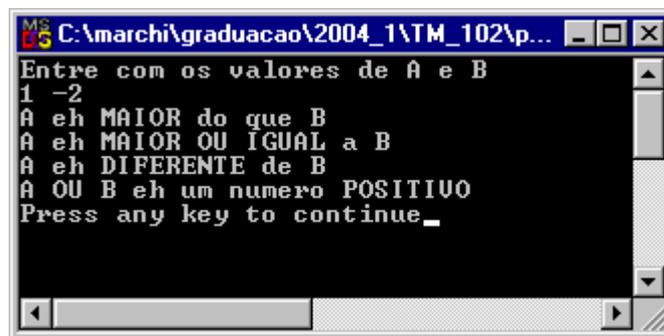
- 9) Até entender, **comparar** os resultados da execução do programa7a.f90, mostrados na Figura 7.1, com cada linha do programa-fonte onde exista um comando IF.
- 10) **Executar** novamente o programa **com A = 1 e B = 1**. Até entender, **comparar** os novos resultados, mostrados na Figura 7.2, com cada linha do programa-fonte onde exista um comando IF.



```
MS-DOS C:\marchi\graduacao\2004_1\TM_102\p...
Entre com os valores de A e B
1 1
A eh MENOR OU IGUAL a B
A eh MAIOR OU IGUAL a B
A eh IGUAL a B
A E B sao numeros POSITIVOS
A OU B eh um numero POSITIVO
Press any key to continue_
```

Figura 7.2 Resultado do programa7a.f90 para $A = 1$ e $B = 1$.

11) **Executar** novamente o programa com $A = 1$ e $B = -2$. Até entender, **comparar** os novos resultados, mostrados na Figura 7.3, com cada linha do programa-fonte onde exista um comando IF.



```
MS-DOS C:\marchi\graduacao\2004_1\TM_102\p...
Entre com os valores de A e B
1 -2
A eh MAIOR do que B
A eh MAIOR OU IGUAL a B
A eh DIFERENTE de B
A OU B eh um numero POSITIVO
Press any key to continue_
```

Figura 7.3 Resultado do programa7a.f90 para $A = 1$ e $B = -2$.

12) **Executar** novamente o programa com outros valores para A e B. Até entender, **comparar** os novos resultados com cada linha do programa-fonte onde exista um comando IF.

7.2 programa7b.f90

- 1) Nesta seção será usado o mesmo projeto da seção anterior. Portanto, deve-se executar o seguinte no Fortran:
 - a) **Clicar** sobre o nome do programa-fonte
 - b) **Edit, Cut** para retirar o programa-fonte do projeto.
- 2) No Fortran, seguindo o [procedimento-padrão](#), **criar** e **inserir** no projeto o programa-fonte **programa7b.f90**
- 3) Dentro do espaço de edição do Fortran, na subjanela maior, **copiar** exatamente o texto em vermelho mostrado na **Tabela 7.5**.
- 4) Comentários sobre o programa:
 - a) Este programa foi escrito para extrair as duas raízes de uma equação do 2º grau.

b) Nesta seção é mostrada e testada a segunda forma de uso do comando IF (SE), cuja sintaxe é apresentada na Tabela 7.6 e usada no programa7b.f90. Esta estrutura do comando IF permite que, se as condições são satisfeitas, sejam executados diversos comandos do programa escritos em várias linhas. Se as condições não são satisfeitas, são executados os comandos que estão entre o comando ELSE e o END IF.

Tabela 7.5 Programa7b.f90.

```
REAL A, B, C
REAL D, X1, X2

WRITE(*,*) "Entre com os valores de A, B e C, onde"
WRITE(*,*) "A*(X**2) + B*X + C = 0"
READ(*,*) A, B, C

D = (B**2) - 4.0 * A * C

IF ( D >= 0.0 ) THEN

    X1 = ( -B + SQRT(D) ) / ( 2.0 * A )
    X2 = ( -B - SQRT(D) ) / ( 2.0 * A )
    WRITE(*,*) "A primeira raiz real eh = ", X1
    WRITE(*,*) "A segunda raiz real eh = ", X2

ELSE

    X1 = -B / ( 2.0 * A )
    X2 = SQRT(ABS(D)) / ( 2.0 * A )
    WRITE(*,*) "A primeira raiz complexa eh = ", X1, " + ", X2, " i"
    WRITE(*,*) "A segunda raiz complexa eh = ", X1, " - ", X2, " i"

END IF

END
```

c) Se não existir necessidade de comandos entre o ELSE e o END IF, pode-se usar a estrutura do comando IF mostrada na Tabela 7.7.

d) Recomenda-se indentar, isto é, deslocar para a direita os comandos entre o IF (SE) e o ELSE (SENÃO), e entre o ELSE e o END IF, para que sejam facilmente visualizados os comandos que são ou não executados. Um exemplo desta recomendação é mostrado no programa7b.f90, na Tabela 7.5.

- e) As três formas de uso do comando IF, mostradas nas Tabelas 7.2, 7.6 e 7.7, podem ser combinadas entre si, cada uma podendo ser inserida dentro das estruturas das outras.

Tabela 7.6 Sintaxe 2 do comando IF.

```
IF ( condições ) THEN
    BLOCO 1 (uma ou várias linhas de programa com comandos)
ELSE
    BLOCO 2 (uma ou várias linhas de programa com comandos)
END IF
```

onde "condições" representa uma ou várias condições que devem ser satisfeitas para que o bloco 1 seja executado;
se as "condições" não forem satisfeitas, o bloco 2 é executado.

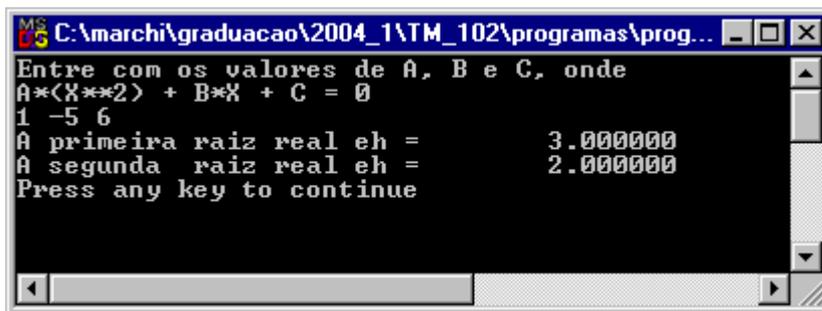
Tabela 7.7 Sintaxe 3 do comando IF.

```
IF ( condições ) THEN
    BLOCO (uma ou várias linhas de programa com comandos)
END IF
```

onde "condições" representa uma ou várias condições que devem ser satisfeitas para que o bloco de linhas de programa seja executado.

- 5) Executar **Build, Compile** para compilar o programa.
- 6) Gerar o programa-executável fazendo **Build, Build**.
- 7) Ao se executar o programa, através de **Build, Execute**, surge uma janela do DOS, mostrada na Figura 7.4, dentro da qual tem-se:
 - a) Na primeira linha, o comentário Entre com os valores de A, B e C, onde, resultado do comando `WRITE(*,*) "Entre com os valores de A, B e C, onde"` do programa.
 - b) Na segunda linha, o comentário $A*(X**2) + B*X + C = 0$, resultado do comando `WRITE(*,*) "A*(X**2) + B*X + C = 0"` do programa.
 - c) Na terceira linha, o programa pára e fica aguardando que sejam fornecidos os valores das variáveis A, B e C, resultado do comando `READ(*,*) A, B, C` do programa. Para que o programa continue sua execução é necessário **digitar 1 -5 6**, por exemplo, e, em seguida, **clique na tecla enter**.

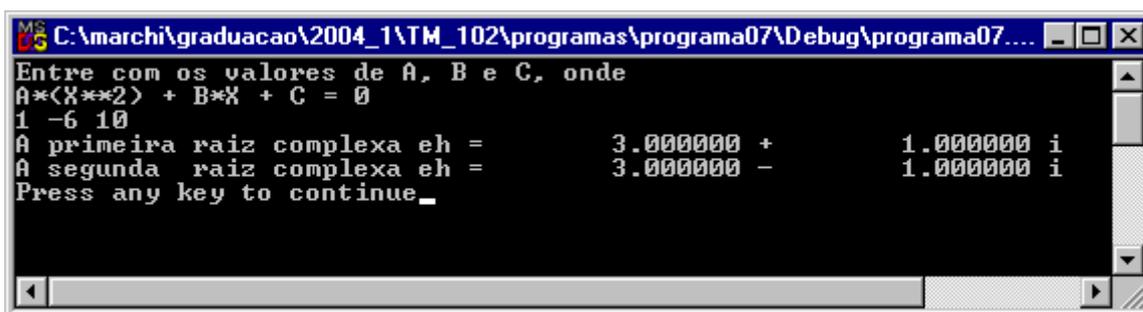
- d) Em seguida são apresentados os resultados correspondentes ao bloco do programa selecionado em função da condição testada no comando IF.



```
MS-DOS C:\marchi\graduacao\2004_1\TM_102\programas\prog...
Entre com os valores de A, B e C, onde
A*(X**2) + B*X + C = 0
1 -5 6
A primeira raiz real eh =          3.000000
A segunda raiz real eh =          2.000000
Press any key to continue
```

Figura 7.4 Resultado do programa7b.f90 para A = 1, B = -5 e C = 6.

- 8) Até entender, **comparar** os resultados da execução do programa7b.f90, mostrados na Figura 7.4, com cada linha do programa-fonte.
- 9) **Executar** novamente o programa com A = 1, B = -6 e C = 10. Até entender, **comparar** os novos resultados, mostrados na Figura 7.5, com cada linha do programa-fonte.



```
MS-DOS C:\marchi\graduacao\2004_1\TM_102\programas\programa07\Debug\programa07....
Entre com os valores de A, B e C, onde
A*(X**2) + B*X + C = 0
1 -6 10
A primeira raiz complexa eh =      3.000000 +      1.000000 i
A segunda raiz complexa eh =      3.000000 -      1.000000 i
Press any key to continue_
```

Figura 7.5 Resultado do programa7b.f90 para A = 1, B = -6 e C = 10.

- 10) **Executar** novamente o programa com outros valores para A, B e C. Até entender, **comparar** os novos resultados com cada linha do programa-fonte.

7.3 programa7c.f90

- 1) Nesta seção será usado o mesmo projeto da seção anterior. Portanto, deve-se executar o seguinte no Fortran:
- a) **Clicar** sobre o nome do programa-fonte
 - b) **Edit, Cut** para retirar o programa-fonte do projeto.
- 2) No Fortran, seguindo o [procedimento-padrão](#), **criar e inserir** no projeto o programa-fonte **programa7c.f90**

- 3) Dentro do espaço de edição do Fortran, na subjanela maior, **copiar** exatamente o texto em vermelho mostrado na **Tabela 7.8**.

Tabela 7.8 Programa7c.f90.

```
INTEGER A

WRITE(*,*) "Entre com o valor inteiro de A"
READ(*,*) A

SELECT CASE ( A )

  CASE ( -1000:1 )
    WRITE(*,*) "Valor de A estah entre -1000 e 1"

  CASE ( 2, 4 )
    WRITE(*,*) "Valor de A eh 2 ou 4"

  CASE ( 3 )
    WRITE(*,*) "Valor de A eh 3"

  CASE ( 5:2000 )
    WRITE(*,*) "Valor de A estah entre 5 e 2000"

  CASE DEFAULT
    WRITE(*,*) "Valor de A eh menor do que -1000 ou maior do que 2000"

END SELECT

END
```

- 4) Comentários sobre o programa:

- a) No programa7c.f90 é usado um novo comando do FORTRAN: SELECT CASE. Este comando é empregado para escolher apenas uma opção entre várias e executar o bloco de linhas ou comandos do programa correspondente à opção escolhida. A opção é feita em função de uma variável do tipo inteiro ou caracter, chamada de variável seletora. Cada bloco pode conter uma ou várias linhas de programa. A sintaxe deste comando é apresentada na Tabela 7.9.
- b) O número de opções existentes depende de cada caso ou programa em particular.
- c) A opção CASE DEFAULT é executada se nenhuma das anteriores for satisfeita.

- d) Recomenda-se indentar, isto é, deslocar para a direita cada opção (CASE) e cada bloco para que sejam facilmente visualizados os comandos ligados a cada opção. Um exemplo desta recomendação é mostrado no programa7c.f90, na Tabela 7.8.
- e) Deve-se notar que em todos os comandos WRITE não é usado o valor da variável A.

Tabela 7.9 Sintaxe do comando SELECT CASE.

```
SELECT CASE ( variavel )
```

```
    CASE ( valor1 )
```

```
        BLOCO 1
```

```
    CASE ( valor2 )
```

```
        BLOCO 2
```

```
    CASE DEFAULT
```

```
        BLOCO DEFAULT
```

```
END SELECT
```

onde "valor" representa um valor inteiro, alguns valores separados por vírgula ou, ainda, um intervalo de valores separados por dois pontos; apenas o bloco de comandos correspondente ao "valor" da variável é executado. A variável de seleção pode ser do tipo inteiro ou caracter. No caso de variável caracter, os "valores" devem ser strings entre aspas.

- 5) Executar **Build, Compile** para compilar o programa.
- 6) Gerar o programa-executável fazendo **Build, Build**.
- 7) Ao se executar o programa, através de **Build, Execute**, surge uma janela, mostrada na Figura 7.6, dentro da qual tem-se:
- a) Na primeira linha, o comentário Entre com o valor inteiro de A, resultado do comando `WRITE(*,*) "Entre com o valor inteiro de A"` do programa.
- b) Na segunda linha, o programa pára e fica aguardando que seja fornecido o valor da variável A, resultado do comando `READ(*,*) A` do programa. Para que o programa continue sua execução é necessário **digitar o valor -10 para a variável A**, por exemplo, e, em seguida, **clique na tecla enter**.

- c) Na terceira linha, é mostrado um comentário correspondente à opção selecionada em função do valor da variável A.
- 8) Até entender, **verificar** se o resultado da execução do programa7c.f90, mostrado na Figura 7.6, está correto em função do valor da variável A e o comentário correspondente à opção selecionada.
- 9) **Executar** novamente o programa **com A = 2**. Até entender, **verificar** se o novo resultado da execução do programa7c.f90 está correto em função do valor da variável A e o comentário correspondente à opção selecionada.
- 10) **Repetir o item 9 com A = 50**.
- 11) **Repetir o item 9 com A = 2345**.
- 12) **Repetir o item 9 com outro valor qualquer**.
- 13) Encerrar a sessão seguindo o [procedimento-padrão](#).

Figura 7.6 Resultado do programa7c.f90 para $A = -10$.

7.4 EXERCÍCIOS

Exercício 7.1

Editar um programa-fonte em FORTRAN para executar o seguinte algoritmo (passos):

- 1) Ler os valores inteiros de três variáveis, denotadas por A, B e C
- 2) Determinar qual das variáveis tem o menor e o maior valor

Exercício 7.2

Alterar o programa7b.f90 para que não ocorra erro de execução quando o usuário entrar com valor nulo para a variável A, e que seja calculado e escrito o valor da única raiz deste caso.

Exercício 7.3

Editar um programa-fonte em FORTRAN para calcular as raízes de uma equação algébrica cúbica.

Exercício 7.4

Editar um programa-fonte em FORTRAN para executar o seguinte algoritmo (passos):

- 1) Mostrar na tela uma lista com pelo menos 4 opções de cálculo: 1 = módulo de A; 2 = raiz quadrada de A; 3 = exponencial de A; 4 = logaritmo de A
- 2) Ler o valor da opção de cálculo desejada
- 3) Ler o valor real da variável A
- 4) Calcular o resultado de A para a opção escolhida pelo usuário
- 5) Mostrar o resultado de A