

OBJETIVOS DO CAPÍTULO

- Conceitos de: condição, operadores relacionais e lógicos, e opção
- Novos comandos do FORTRAN a usar:
 - IF – THEN – ELSE – END IF
 - SELECT CASE – CASE – CASE DEFAULT – END SELECT

7.1 programa7a.f90

- 1) No Fortran, **criar um projeto** com o nome **programa07**
- 2) No Fortran, **criar e inserir** no projeto o programa-fonte **programa7a.f90**
- 3) No Fortran, **copiar** exatamente o texto em vermelho mostrado na **Tabela 7.1**.
- 4) 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 executar comandos com base em condições. Se as condições são satisfeitas, executa-se o comando que está na mesma linha do IF. Caso as condições não sejam satisfeitas, o comando não é executado.
 - 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, como mostrados em outros exemplos com IF neste programa.
 - e) Nas demais linhas do programa, são testados todos os operadores relacionais e lógicos.
- 5) Executar **Build, Compile** para compilar o programa.
- 6) Gerar o programa-executável com **Build, Build**.
- 7) 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 reais de A e B”, resultado do comando **WRITE(*,*) "Entre com os valores reais de A e B"** do programa.

Tabela 7.1 Programa7a.f90.

```

REAL A, B

WRITE(*,*) "Entre com os valores reais 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"

if ( 1 < B*3 ) write(*,*) "1 eh menor que B*3"

if ( a+2 < b/3 ) write(*,*) "a+2 < b/3"

if ( 1>b .and. a>0 .and. a/=b ) write(*,*) "1>b .and. a>0 .and. a/=b"

END

```

- b) Na segunda linha, o programa pára e fica aguardando até que sejam fornecidos os valores das variáveis A e B, resultado do comando `READ(*,*) A, B` do programa. Para que o programa continue a sua execução, é necessário **digitar 1 2**, por exemplo, e, em seguida, **clicar 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.
- 8) 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.
- 9) **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.

Tabela 7.2 Sintaxe mais simples do comando IF.

IF (condições) COMANDO

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

Tabela 7.3 Operadores relacionais do FORTRAN.

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

Tabela 7.4 Alguns operadores lógicos do FORTRAN.

Operador	Significado
.AND.	e
.OR.	ou

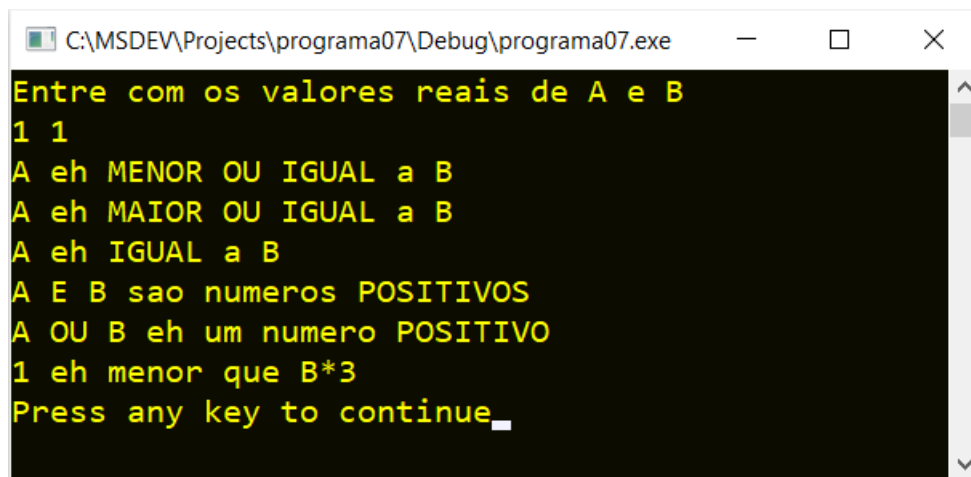
```

C:\MSDEV\Projects\programa07\Debug\programa07.exe
Entre com os valores reais de A e B
1 2
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
1 eh menor que B*3
Press any key to continue_
  
```

Figura 7.1 Resultado do programa7a.f90 para A = 1 e B = 2.

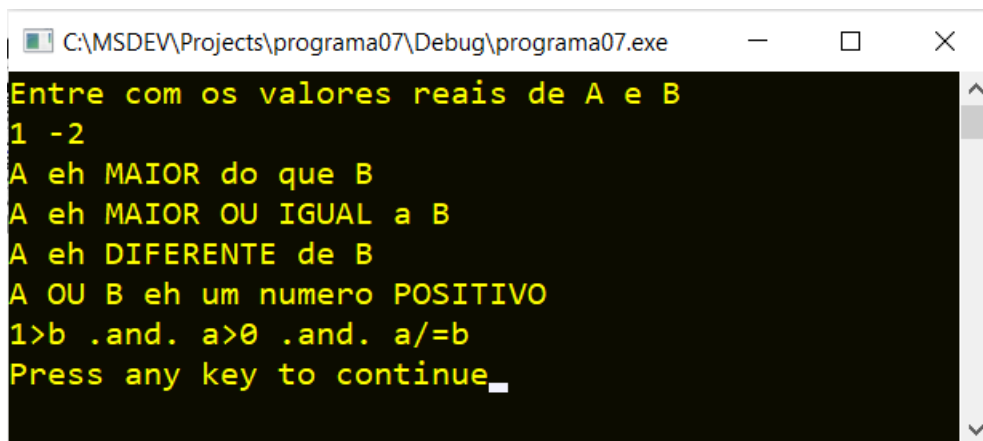
- 10) **Executar** novamente o programa com **A = 1 e B = -2**. Até entender, **comparar** os novos resultados, mostrados na Figura 7.3, com cada linha do programa-fonte onde exista um comando IF.
- 11) **Executar** novamente o programa com **A = -1 e B = -2**. Até entender, **comparar** os novos resultados, mostrados na Figura 7.4, com cada linha do programa-fonte onde exista um comando IF.

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



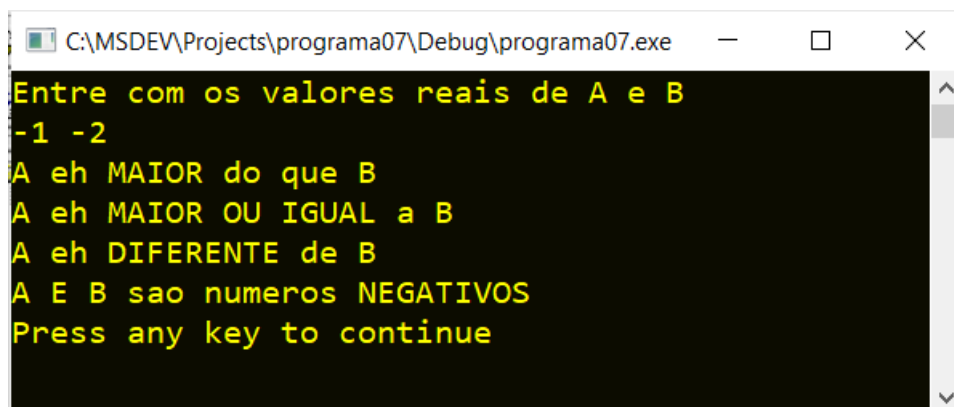
```
C:\MSDEV\Projects\programa07\Debug\programa07.exe
Entre com os valores reais de A e B
1 1
A eh MENOR OU IGUAL a B
A eh MAIOR OU IGUAL a B
A eh IGUAL a B
A E B sao numeros POSITIVOS
A OU B eh um numero POSITIVO
1 eh menor que B*3
Press any key to continue_
```

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



```
C:\MSDEV\Projects\programa07\Debug\programa07.exe
Entre com os valores reais 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
1>b .and. a>0 .and. a/=b
Press any key to continue_
```

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



```
C:\MSDEV\Projects\programa07\Debug\programa07.exe
Entre com os valores reais 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 E B sao numeros NEGATIVOS
Press any key to continue_
```

Figura 7.4 Resultado do programa7a.f90 para A = -1 e B = -2.

7.2 programa7b.f90

- 1) No Fortran, **criar e inserir** no projeto o programa-fonte **programa7b.f90**
- 2) No Fortran, **copiar** exatamente o texto em vermelho mostrado na **Tabela 7.5**.

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

write(*,*) "Determinante =", D

if ( A == 0 ) then
    write(*,*) "Voce entrou com A = 0."
    x1 = - c / b
    write(*,*) "A unica raiz real eh = ", x1
end if

IF ( D >= 0.0 .and. a /= 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

    IF ( a /= 0.0 ) THEN
        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 IF

END
```

3) Comentários sobre o programa:

- a) Este programa foi escrito para extrair as duas raízes de uma equação do 2º grau, e a única raiz no caso da equação ser linear.
- b) Nesta seção são mostradas e testadas as duas outras formas de uso do comando IF (SE), cujas sintaxes são apresentadas nas Tabelas 7.6 e 7.7, e usadas no programa7b.f90.
- c) Na estrutura mais completa do comando IF (Tabela 7.6), se as condições são satisfeitas, são executados diversos comandos do programa que estão escritos em várias linhas entre IF e ELSE. Se as condições não são satisfeitas, são executados os comandos que estão entre ELSE e END IF.
- d) 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, que também é exemplificada no programa7b.f90.
- e) Pode-se colocar um IF dentro de outro, conforme exemplificado no programa7b.f90.
- f) 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 pertencem aos blocos do IF. Um exemplo desta recomendação é mostrado no programa7b.f90, na Tabela 7.5.
- g) 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 mais completa 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
```

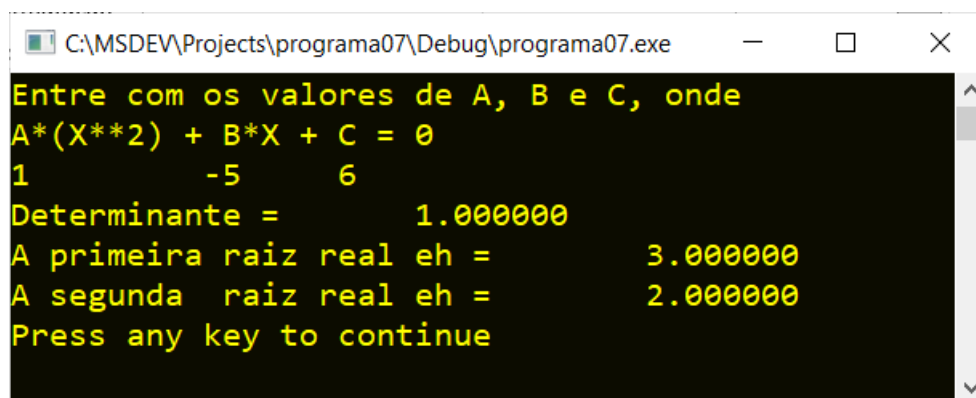
onde "condições" representa uma ou várias condições que devem ser satisfeitas para que o BLOCO 1 seja executado;
se as "condições" não forem satisfeitas, o BLOCO 2 é executado.

Tabela 7.7 Sintaxe intermediária 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.

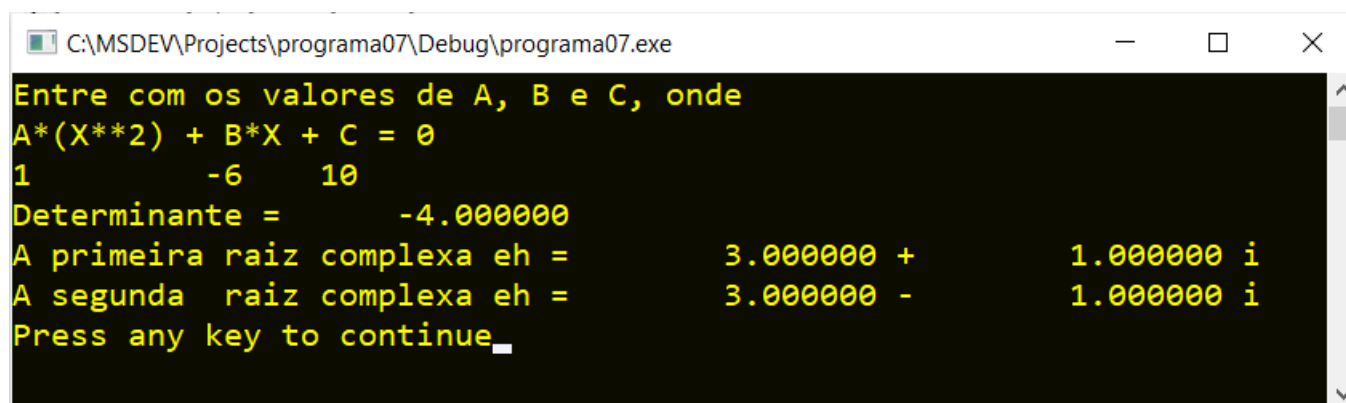
- 4) Executar **Build, Compile** para compilar o programa.
- 5) Gerar o programa-executável com **Build, Build**.
- 6) Ao se executar o programa, através de **Build, Execute**, surge uma janela do DOS, mostrada na Figura 7.5, dentro da qual tem-se:
 - a) Na primeira linha, o comentário “Entre com os 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 até 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 a sua execução, é necessário **digitar 1 -5 6**, por exemplo, e, em seguida, **clicar na tecla enter**.
 - d) Em seguida são apresentados os resultados correspondentes ao bloco do programa selecionado em função da condição testada no comando IF.



```
C:\MSDEV\Projects\programa07\Debug\programa07.exe
Entre com os valores de A, B e C, onde
A*(X**2) + B*X + C = 0
1      -5      6
Determinante =      1.000000
A primeira raiz real eh =      3.000000
A segunda raiz real eh =      2.000000
Press any key to continue
```

Figura 7.5 Resultado do programa7b.f90 para A = 1, B = -5 e C = 6.

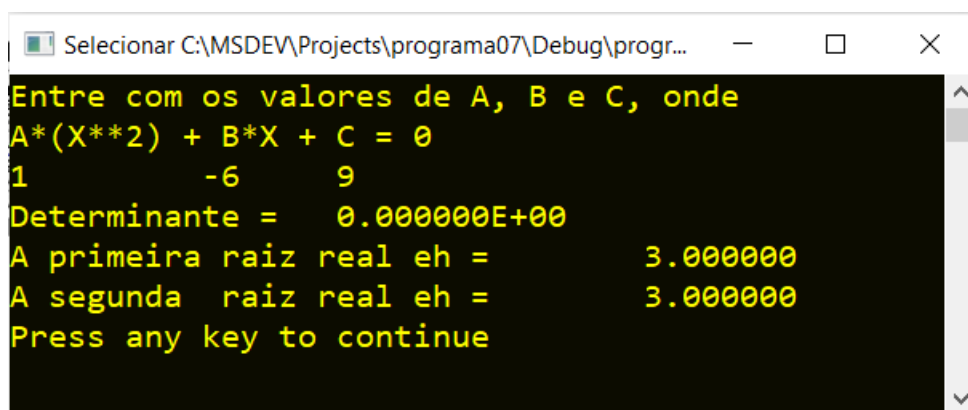
- 7) Até entender, **comparar** os resultados da execução do programa7b.f90, mostrados na Figura 7.5, com cada linha do programa-fonte.
- 8) **Executar** novamente o programa com A = 1, B = -6 e C = 10. Até entender, **comparar** os novos resultados, mostrados na Figura 7.6, com cada linha do programa-fonte.



```
C:\MSDEV\Projects\programa07\Debug\programa07.exe
Entre com os valores de A, B e C, onde
A*(X**2) + B*X + C = 0
1      -6     10
Determinante =     -4.000000
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 7.6 Resultado do programa7b.f90 para A = 1, B = -6 e C = 10.

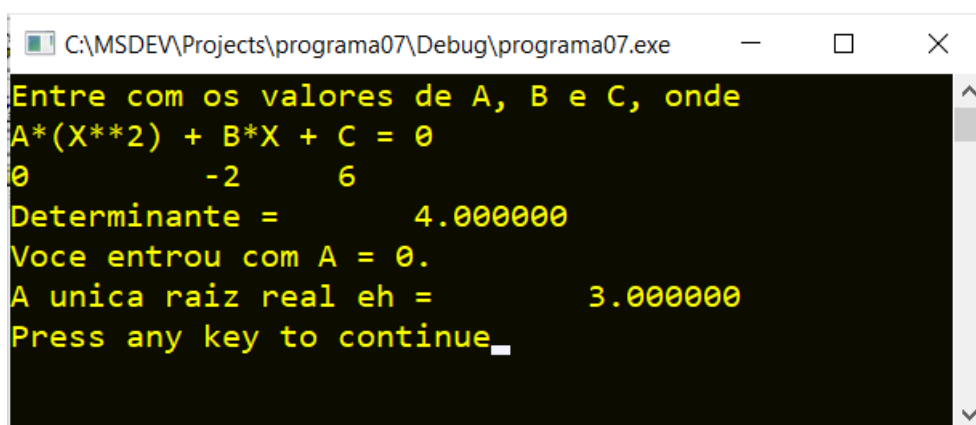
- 9) **Executar** novamente o programa com **A = 1, B = -6 e C = 9**. Até entender, **comparar** os novos resultados, mostrados na Figura 7.7, com cada linha do programa-fonte.



```
Selecionar C:\MSDEV\Projects\programa07\Debug\progr...
Entre com os valores de A, B e C, onde
A*(X**2) + B*X + C = 0
1          -6          9
Determinante = 0.000000E+00
A primeira raiz real eh = 3.000000
A segunda raiz real eh = 3.000000
Press any key to continue
```

Figura 7.7 Resultado do programa7b.f90 para A = 1, B = -6 e C = 9.

- 10) **Executar** novamente o programa com **A = 0, B = -2 e C = 6**. Até entender, **comparar** os novos resultados, mostrados na Figura 7.8, com cada linha do programa-fonte.



```
C:\MSDEV\Projects\programa07\Debug\programa07.exe
Entre com os valores de A, B e C, onde
A*(X**2) + B*X + C = 0
0          -2          6
Determinante = 4.000000
Voce entrou com A = 0.
A unica raiz real eh = 3.000000
Press any key to continue_
```

Figura 7.8 Resultado do programa7b.f90 para A = 0, B = -2 e C = 6.

- 11) **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) No Fortran, **criar e inserir** no projeto o programa-fonte **programa7c.f90**
- 2) No Fortran, **copiar** exatamente o texto em vermelho mostrado na **Tabela 7.8**.
- 3) 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.

Tabela 7.8 Programa7c.f90.

```
INTEGER A

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

SELECT CASE ( A )

    CASE ( -1000:1 )
        WRITE(*,*) "-1000 <= A <= 1"

    CASE ( 2, 4 )
        WRITE(*,*) "A = 2 ou 4"

    CASE ( 3 )
        WRITE(*,*) "A = 3"

    CASE ( 5:2000 )
        WRITE(*,*) " 5 <= A <= 2000"

    CASE ( : -1001 )
        WRITE(*,*) " A <= -1001"

    CASE ( 3000 : )
        WRITE(*,*) " A >= 3000"

    CASE DEFAULT ! não é obrigatório
        WRITE(*,*) "2001 <= A <= 2999"

END SELECT

END
```

- b) O número de opções existentes depende de cada caso ou programa em particular.
- c) Não pode existir sobreposição entre todas as opções.
- d) A opção CASE DEFAULT é executada se nenhuma das anteriores for selecionada. Ela é opcional.
- e) Pode-se usar intervalo aberto de um lado. Um exemplo é a linha **CASE (3000 :)** do programa.

- f) Recomenda-se indentar, isto é, deslocar para a direita cada opção (CASE) e cada bloco para que sejam facilmente visualizados os comandos de cada opção. Um exemplo desta recomendação é mostrado no programa7c.f90, na Tabela 7.8.
- g) 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 ( caso )
```

```
    CASE ( caso1 )
```

```
        BLOCO 1
```

```
    CASE ( caso2 )
```

```
        BLOCO 2
```

```
    CASE DEFAULT
```

```
        BLOCO DEFAULT
```

```
END SELECT
```

onde "caso1" e "caso2" representam valores inteiros: alguns valores separados por vírgula; um intervalo definido de valores separados por dois pontos; um intervalo aberto de um lado; ou combinações das anteriores.

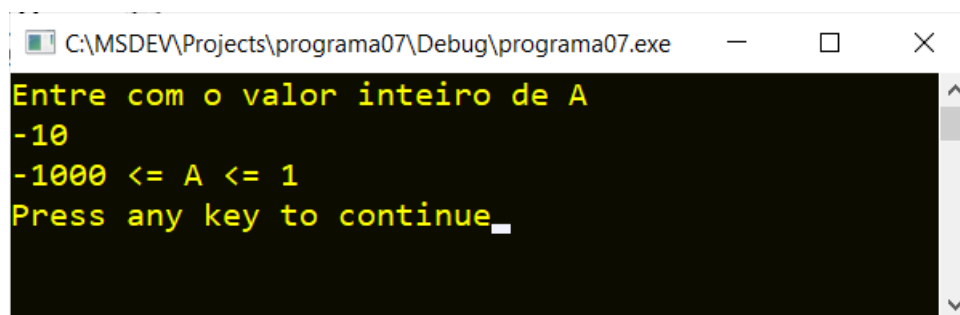
Apenas o bloco de comandos correspondente ao "caso" da variável é executado.

A variável de seleção pode ser do tipo inteiro ou caracter.

No caso de variável caracter, os "casos" devem ser strings entre aspas.

- 4) Executar **Build, Compile** para compilar o programa.
- 5) Gerar o programa-executável com **Build, Build**.
- 6) Ao se executar o programa, através de **Build, Execute**, surge uma janela, mostrada na Figura 7.9, 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 até que seja fornecido o valor da variável A, resultado do comando `READ(*,*) A` do programa. Para que o programa continue a sua execução, é necessário **digitar o valor -10 para a variável A**, por exemplo, e, em seguida, **clicar na tecla enter**.
 - c) Na terceira linha, é mostrado um comentário correspondente à opção selecionada em função do valor da variável A.
- 7) Até entender, **verificar** se o resultado da execução do programa7c.f90, mostrado na Figura 7.9, está correto em função do valor da variável A e o comentário correspondente à opção selecionada.

- 8) **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.
- 9) **Repetir o item 8 com A = 3**.
- 10) **Repetir o item 8 com A = 50**.
- 11) **Repetir o item 8 com A = -5000**.
- 12) **Repetir o item 8 com A = 4000**.
- 13) **Repetir o item 8 com A = 2500**.
- 14) **Repetir o item 8 com outro valor qualquer**.



```
C:\MSDEV\Projects\programa07\Debug\programa07.exe
Entre com o valor inteiro de A
-10
-1000 <= A <= 1
Press any key to continue_
```

Figura 7.9 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 três variáveis tem o menor e o maior valor.
- 3) Escrever em arquivo os resultados do item 2 junto com os valores das variáveis.
- 4) Abrir automaticamente o arquivo com o NOTEPAD.
- 5) Usar A = 2, B = 7 e C = -1.

Exercício 7.2

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

Exercício 7.3

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 decimal 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

Exercício 7.4

Editar um programa-fonte em FORTRAN para calcular o mínimo múltiplo comum entre dois números inteiros informados pelo usuário

Exercício 7.5

Editar um programa-fonte em FORTRAN para calcular o mínimo múltiplo comum entre três números inteiros informados pelo usuário

Exercício 7.6

Editar um programa-fonte em FORTRAN para calcular, a partir de 5 números inteiros informados pelo usuário:

- 1) a soma dos números positivos
- 2) a soma dos números negativos
- 3) a média aritmética dos 5 números

Testar com 1, 2, 3, -1 e -2

Exercício 7.7

- 1) Criar um projeto com o nome Cap7 e inserir nele o programa-fonte Cap7.f90
- 2) Ler os valores inteiros de três números.
- 3) Criar o arquivo de saída Cap7.TXT e escrever nele o nome completo do aluno.
- 4) Usando o comando IF, encontrar o maior número e atribuí-lo à variável Max.
- 5) Escrever no arquivo de saída o valor de Max junto com o seu nome
- 6) Abrir o arquivo Cap7.TXT com o aplicativo Bloco de Notas (Notepad).
- 7) Executar o programa com: 1, 2 e 3

O resultado esperado é: Max = 3

Exercício 7.8

- 1) Criar um projeto com o nome Cap7b e inserir nele o programa-fonte Cap7b.f90
- 2) Ler o valor inteiro da variável V.
- 3) Criar o arquivo de saída Cap7b.TXT e escrever nele:
 - a) o nome completo do aluno
 - b) o valor da variável V e seu nome
- 4) Usando o comando SELECT CASE, escrever no arquivo de saída um comentário indicando se o valor de V é positivo, negativo ou nulo.
- 5) Abrir o arquivo de saída com o aplicativo Bloco de Notas (Notepad).