

OBJETIVOS DO CAPÍTULO

- Conceitos de: variáveis do tipo real, tradução de expressões algébricas em FORTRAN, mistura de variáveis do tipo inteiro com real, prioridade nas operações matemáticas, dígitos significativos, notações decimal e científica
- Operadores matemáticos básicos com números reais
- Novos comando do FORTRAN a usar: REAL

Para inicializar as atividades deste capítulo, deve-se executar o programa Fortran.

3.1 programa03c.f90

- 1) Neste capítulo será usado um projeto já existente, no caso o projeto **programa03** do fim do capítulo 2. Para isso, deve-se executar o seguinte no Fortran:
 - a) **File, Open Workspace**
 - b) **Indicar** a pasta do projeto chamado **programa03**
 - c) **Selecionar** o arquivo que tem o nome do projeto e extensão mdp, no caso **programa03.mdp**
 - d) Na subjanela do lado esquerdo do Fortran, **clicar** sobre o sinal + ao lado do nome do projeto; deverá aparecer o nome do último programa-fonte que foi inserido neste projeto.
 - e) **Clicar** sobre o nome do programa-fonte para ele ficar realçado.
 - f) **Edit, Cut** para retirar a referência deste programa-fonte do projeto. Este comando não deletará o programa-fonte. Ele apenas o retirará do projeto. Quando for necessário, ele poderá ser inserido no projeto novamente, bastando para isso selecionar o programa-fonte desejado dentro da sua pasta e usar o comando Insert, Files into Project.
- 2) No Fortran, seguindo o procedimento do Capítulo 1, **criar** e **inserir** no projeto o programa-fonte **programa03c.f90**
- 3) **Clicar duas vezes sobre o nome do programa-fonte** para que seja aberta a sua subjanela de edição.
- 4) Conforme é mostrado na Figura 3.1, dentro do espaço de edição do Fortran, na subjanela maior, **copiar** exatamente o texto mostrado na **Tabela 3.1**, abaixo, que está em vermelho.
- 5) Comentários sobre o programa:
 - a) Nos capítulos 1 e 2 foram usados os comandos WRITE, END, INTEGER e READ da linguagem FORTRAN. No programa03c.f90 há um novo comando: **REAL**. Ele é usado para definir variáveis do tipo real, isto é, variáveis que podem guardar ou armazenar na memória do computador números reais, positivos ou negativos, como 1.0, -1.0, 1.1, 3.1416 ou -0.003.

- b) Deve-se lembrar que em português se usa vírgula para separar a parte inteira da decimal de um número real. Mas, em FORTRAN, é usado o sinal de ponto com este fim.
- c) A linha **REAL A** define a variável A como sendo do tipo real. Este comando reserva um espaço na memória do computador, utilizando o nome ou rótulo A para armazenar um valor real. As demais linhas do programa já foram comentadas no capítulo 2.
- d) Uma variável real definida como na Tabela 3.1 é chamada de real simples ou de precisão simples.
- e) São usados 6 dígitos significativos para representar uma variável real de precisão simples. O seu menor valor real é aproximadamente 1.17×10^{-38} , e o seu maior valor é 3.40×10^{38} .
- 6) Executar **Build, Compile** para compilar o programa
- 7) Gerar o programa-executável fazendo **Build, Build**.

Tabela 3.1 Programa03c.f90.

```
REAL A
WRITE(*,*) "Entre com o valor de A"
READ(*,*) A
WRITE(*,*) "Valor de A = ", A
END
```

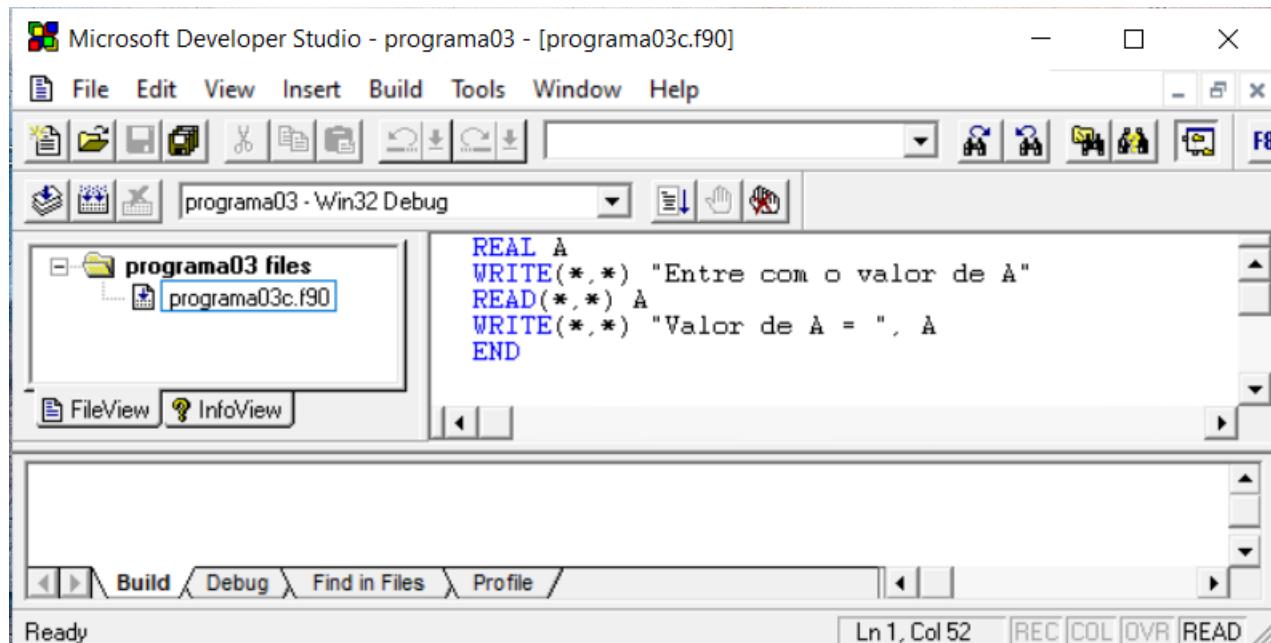


Figura 3.1 Programa03c.f90.

- 8) Ao se executar o programa, através de **Build, Execute**, surge uma janela, mostrada na Figura 3.2, dentro da qual tem-se:
- Na primeira linha, o comentário “Entre com o valor de A”, resultado do comando

```
WRITE(*,*) "Entre com o valor 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 1 para a variável A**, por exemplo, e, em seguida, **clicar na tecla enter**.
- c) Na terceira linha, o comentário “Valor de A =” e o valor da variável A, resultado do comando `WRITE(*,*) "Valor de A = ", A` do programa. Deve-se notar que foi fornecido um valor inteiro para a variável A, no caso 1. Mas o resultado do programa mostra o valor 1.000000 porque a variável A é do tipo real.
- d) Na quarta linha, a frase “Press any key to continue”. Como diz este aviso, basta clicar em qualquer tecla para continuar. Ao se fazer isso, a execução do programa é encerrada.

```
C:\MSDEV\Projects\programa03\Debug\programa03.exe
Entre com o valor de A
1
Valor de A = 1.000000
Press any key to continue
```

Figura 3.2 Resultado da execução do programa03c.f90 com A = 1.

- 9) Executar novamente o programa, através de **Build, Execute**, com A = -0.0031; este número está escrito na chamada notação decimal. Junto a um número real, a letra E é usada para representar números através da notação científica ou exponencial. Pode-se fornecer dados ao programa usando tanto a notação decimal quanto a científica. Desta forma, o valor -3.100000E-03 da variável A, mostrado na Figura 3.3, significa -3.1×10^{-3} , que é igual a -0.0031.

```
C:\MSDEV\Projects\programa03\Debug\programa03.exe
Entre com o valor de A
-0.0031
Valor de A = -3.100000E-03
Press any key to continue
```

Figura 3.3 Resultado da execução do programa03c.f90 com A = -0.0031.

- 10) Executar novamente o programa, através de **Build, Execute**, com A = 1.0E+2, como no exemplo mostrado na Figura 3.4. São aceitas diversas formas de se entrar com o mesmo número. Por exemplo, A = 1.0E+2 = 100 = 1.0E2 = 1E2 = 1.E2 = 1+2. Mas não é permitido A = 1.0x10².
- 11) No caso de números menores que 1, pode-se entrar no FORTRAN com, por exemplo: A = 0.01 = 1.0E-2 = 1.E-2 = 1-2. Mas não é permitido A = 1.0x10⁻².

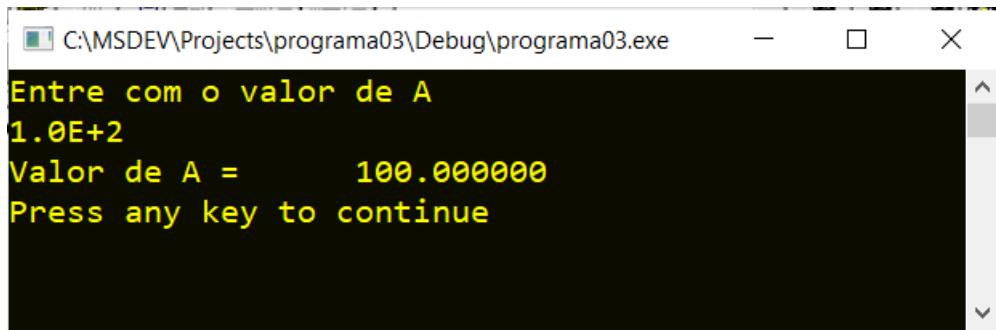


Figura 3.4 Resultado da execução do programa03c.f90 com A = 1.0E+2.

3.2 programa03d.f90

- 1) Nesta seção será usado um projeto já existente, no caso o 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 a referência do programa-fonte do projeto.
- 2) No Fortran, seguindo o procedimento do Capítulo 1, **criar** e **inserir** no projeto o programa-fonte **programa03d.f90**
- 3) **Clicar duas vezes sobre o nome do programa-fonte** para que seja aberta a sua subjanela de edição.
- 4) Conforme é mostrado na Figura 3.1, dentro do espaço de edição do Fortran, na subjanela maior, **copiar** exatamente o texto mostrado na **Tabela 3.2**, abaixo, que está em vermelho.

Tabela 3.2 Programa03d.f90.

```

INTEGER D
REAL A, B, C

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

D = A
B = 1 / D
C = 1.0 / D

WRITE(*,*) "A = ", A
WRITE(*,*) "D = ", D
WRITE(*,*) "B = ", B
WRITE(*,*) "C = ", C

END

```

- 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 3.5, dentro da qual tem-se:
 - a) Na primeira linha, o comentário “Entre com o valor REAL de A”, resultado do comando `WRITE(*,*) "Entre com o valor REAL 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 2.9 para a variável A**, por exemplo, e, em seguida, **clicar na tecla enter**.
 - c) Nas linhas seguintes, são mostrados os valores das variáveis A, D, B e C, respectivamente.

```

Selecionar C:\MSDEV\Projects\programa03\Debug\programa03d.f90
Entre com o valor REAL de A
2.9
A = 2.900000
D = 2
B = 0.000000E+00
C = 5.000000E-01
Press any key to continue
  
```

Figura 3.5 Resultado da execução do programa03d.f90.

- 8) Comentários sobre os resultados do programa03d.f90 mostrados na Figura 3.5:
 - a) Na linha `D = A` do programa, o valor da variável real A é atribuído à variável inteira D. Portanto, apenas a parte inteira do número A é passada à variável D, desprezando-se a parte decimal.
 - b) Na linha `B = 1 / D` do programa, o resultado da divisão do número 1 (inteiro) pela variável inteira D é atribuído à variável real B. Conforme visto no capítulo 2, o valor que resulta de um cálculo envolvendo dois números inteiros também é um número inteiro, que corresponde à parte inteira do número real equivalente ao cálculo realizado. Portanto, o resultado de 1 dividido por 2 (variável D) resulta em 0.5, que é um número real. Mas como o cálculo envolve dois números inteiros, a parte inteira do número real 0.5 é 0, que é o resultado da variável B mostrado na Figura 3.5. Assim, apenas a parte inteira da divisão 1 / D é passada à variável real B, desprezando-se a parte decimal. Enfatiza-se que ocorre um truncamento e não um arredondamento.
 - c) Na linha `C = 1.0 / D` do programa, o resultado da divisão do número 1.0 (real) pela variável inteira D é atribuído à variável real C. Neste caso, o cálculo envolve a divisão de um número real (1.0) por um número inteiro (D). O resultado da variável C mostrado na Figura 3.5 é igual a 0.5, que é um número real. Isso ocorre porque o valor que resulta de um cálculo envolvendo um número inteiro e um número real é um número real.

3.3 programa03e.f90

- 1) Nesta seção será usado um projeto já existente, no caso o 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 a referência do programa-fonte do projeto.
- 2) No Fortran, seguindo o procedimento do Capítulo 1, **criar e inserir** no projeto o programa-fonte **programa03e.f90**
- 3) **Clicar duas vezes sobre o nome do programa-fonte** para que seja aberta a sua subjanela de edição.
- 4) Dentro do espaço de edição do Fortran, na subjanela maior, **copiar** exatamente o texto mostrado na **Tabela 3.3**, abaixo, que está em vermelho.

Tabela 3.3 Programa03e.f90.

```
REAL A, B, C, D, E, F, G

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

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

C = A + B
D = B - A
E = A * B
F = A / B
G = A ** B

WRITE(*,*) "A = ", A
WRITE(*,*) "B = ", B
WRITE(*,*) "C = A + B = ", C
WRITE(*,*) "D = B - A = ", D
WRITE(*,*) "E = A * B = ", E
WRITE(*,*) "F = A / B = ", F
WRITE(*,*) "G = A ** B = ", G

END
```

- 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 3.6, dentro da qual tem-se:

a) Na primeira linha, o comentário “Entre com o valor de A”, resultado do comando

`WRITE(*,*) "Entre com o valor de A"` do programa.

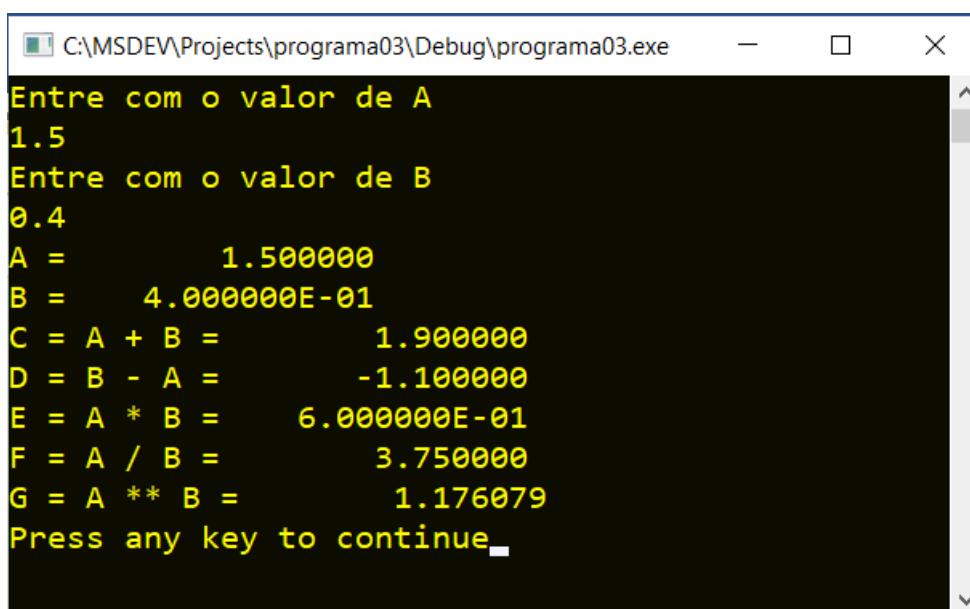
b) Na segunda linha, o programa pára e fica aguardando que seja fornecido o valor da variável A, resultado do comando `READ(*,*) A` do programa. Para que o programa continue sua execução é necessário **digitar o valor 1.5 para a variável A**, por exemplo, e, em seguida, **clicar na tecla enter**.

c) Na terceira linha, o comentário “Entre com o valor de B”, resultado do comando

`WRITE(*,*) "Entre com o valor de B"` do programa.

d) Na quarta linha, o programa pára e fica aguardando que seja fornecido o valor da variável B, resultado do comando `READ(*,*) B` do programa. Para que o programa continue sua execução é necessário **digitar o valor 0.4 para a variável B**, por exemplo, e, em seguida, **clicar na tecla enter**.

e) Nas linhas seguintes, são mostrados os valores de todas as variáveis.



```
Entre com o valor de A
1.5
Entre com o valor de B
0.4
A =      1.500000
B =      4.000000E-01
C = A + B =      1.900000
D = B - A =      -1.100000
E = A * B =      6.000000E-01
F = A / B =      3.750000
G = A ** B =      1.176079
Press any key to continue.
```

Figura 3.6 Resultado da execução do programa03e.f90.

- 8) **Analizar** cada resultado mostrado na Figura 3.6 comparando-o com o valor esperado obtido de um cálculo mental ou com o uso de uma calculadora.
- 9) **Executar** novamente o programa com outros valores para A e B. Analisar cada novo resultado comparando-o com o valor esperado obtido de um cálculo mental ou com o uso de uma calculadora.
 - a) Exemplo 2: usar **A = 0** e **B = 0.4**.
 - b) Exemplo 3: usar **A = 1.5** e **B = 0**. Neste caso, haverá uma divisão por zero no cálculo da variável F e, este compilador, irá parar a execução do programa; portanto, deve-se evitar divisões por zero. Em geral, compiladores modernos não param a execução mas geram não-números (NaN).

3.4 ESCALA DE PRIORIDADES ENTRE OS OPERADORES MATEMÁTICOS

Na linguagem FORTRAN, as operações matemáticas são executadas em uma sequência lógica de acordo com uma escala de prioridades entre elas, que é apresentada na Tabela 3.4. Além disso são consideradas as regras descritas da Tabela 3.5. É importante conhecer estas prioridades e regras para se fazer corretamente a tradução de expressões algébricas em expressões na linguagem FORTRAN, dentro de um programa-fonte. Alguns exemplos de expressões algébricas são apresentados abaixo nas Equações (3.1) a (3.9).

Tabela 3.4 Escala de prioridades entre os operadores matemáticos básicos em FORTRAN.

Prioridade	Operação	Símbolo	Nome do símbolo
1 ^a	Qualquer	()	Parênteses
2 ^a	Potenciação	**	Duplo asterisco
3 ^a	Multiplicação ou Divisão	*	Asterisco
		/	Barra
4 ^a	Adição ou Subtração	+	Sinal mais
		-	Sinal menos

Tabela 3.5 Regras adotadas nas operações matemáticas em FORTRAN.

Regra	Descrição
1	No caso de operações matemáticas que têm a mesma prioridade (multiplicação e divisão ou adição e subtração), executa-se primeiro o cálculo mais à esquerda
2	Parênteses são usados para que os cálculos sejam executados na ordem que se deseja. Com eles, a ordem de execução é do parênteses mais interno para o mais externo, da esquerda para a direita.
3	Com duas ou mais potenciações consecutivas, a ordem de execução é da direita para a esquerda.
4	O valor que resulta de um cálculo envolvendo dois números inteiros também é um número inteiro, que corresponde à parte inteira do número real equivalente ao cálculo realizado.
5	Uma operação envolvendo um número real com um número inteiro resulta em um número real.
6	A atribuição de um número real a uma variável do tipo inteiro resulta em um número inteiro que trunca a parte decimal do número real.
7	A atribuição de um número inteiro a uma variável do tipo real resulta em um número real.
8	Quando se somam ou se subtraem dois números com magnitudes muito diferentes, é possível que o valor menor seja desprezado; a isso se denomina de erro de arredondamento ou de truncamento.
9	Espaços em branco entre variáveis, sinais de igualdade, parênteses, números ou operadores matemáticos não interferem no resultado dos cálculos. Eles têm apenas função estética.
10	Não pode haver cruzamento entre parênteses: o interno tem que abrir e fechar dentro do externo.

$$H_1 = A + B - C \quad (3.1)$$

$$H_2 = A + \frac{B}{C} \quad (3.2)$$

$$H_3 = \frac{A + B}{C} \quad (3.3)$$

$$H_4 = A + \frac{B}{C} D \quad (3.4)$$

$$H_5 = A + \frac{B}{C D} \quad (3.5)$$

$$H_6 = A + B C^D = H_7 \quad (3.6)$$

$$H_8 = 1 \times 10^{10} + 1 \times 10^{20} \quad (3.7)$$

$$H_9 = C^{3^2} \quad (3.8)$$

$$H_{10} = \frac{1}{3} \sqrt{B^3} = H_{11} = H_{12} \quad (3.9)$$

3.5 programa03f.f90

- 1) Nesta seção será usado um projeto já existente, no caso o projeto da seção 3.3. Portanto, deve-se executar o seguinte no Fortran:
 - a) **Clicar** sobre o nome do programa-fonte
 - b) **Edit, Cut** para retirar a referência do programa-fonte do projeto.
- 2) No Fortran, seguindo o procedimento do Capítulo 1, **criar e inserir** no projeto o programa-fonte **programa03f.f90**
- 3) **Clicar duas vezes sobre o nome do programa-fonte** para que seja aberta a sua subjanela de edição.
- 4) Dentro do espaço de edição do Fortran, na subjanela maior, **copiar** exatamente o texto mostrado na **Tabela 3.6**, abaixo, que está em vermelho.

Tabela 3.6 Programa03f.f90.

```

REAL A, C, B, D
REAL H1, H2, H3, H4, H5, H6, H7, H8, H9, H10, H11, H12

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

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

```

```

WRITE(*,*) "B = ", B
WRITE(*,*) "C = ", C
WRITE(*,*) "D = ", D

WRITE(*,*)

H1 = A + B - C
H2 = A + B / C
H3 = (A + B) / C

H4 = A + B / C * D
H5 = A + B / (C * D)

H6 = A + B * C ** D
H7 = A + (B * (C ** D))

H8 = 1.0E10 + 1.0E20

H9 = C ** 3 ** 2

H10 = (1.0 / 3) * ((B**3) ** (1 / 2.0))
H11 = (1.0 / 3.0) * ((B**3) ** (1.0 / 2.0))
H12 = (1 / 3) * ((B**3) ** (1 / 2))

WRITE(*,*) "H1 = ", H1
WRITE(*,*) "H2 = ", H2
WRITE(*,*) "H3 = ", H3
WRITE(*,*) "H4 = ", H4
WRITE(*,*) "H5 = ", H5
WRITE(*,*) "H6 = ", H6
WRITE(*,*) "H7 = ", H7
WRITE(*,*) "H8 = ", H8
WRITE(*,*) "H9 = ", H9
WRITE(*,*) "H10 = ", H10
WRITE(*,*) "H11 = ", H11
WRITE(*,*) "H12 = ", H12

END

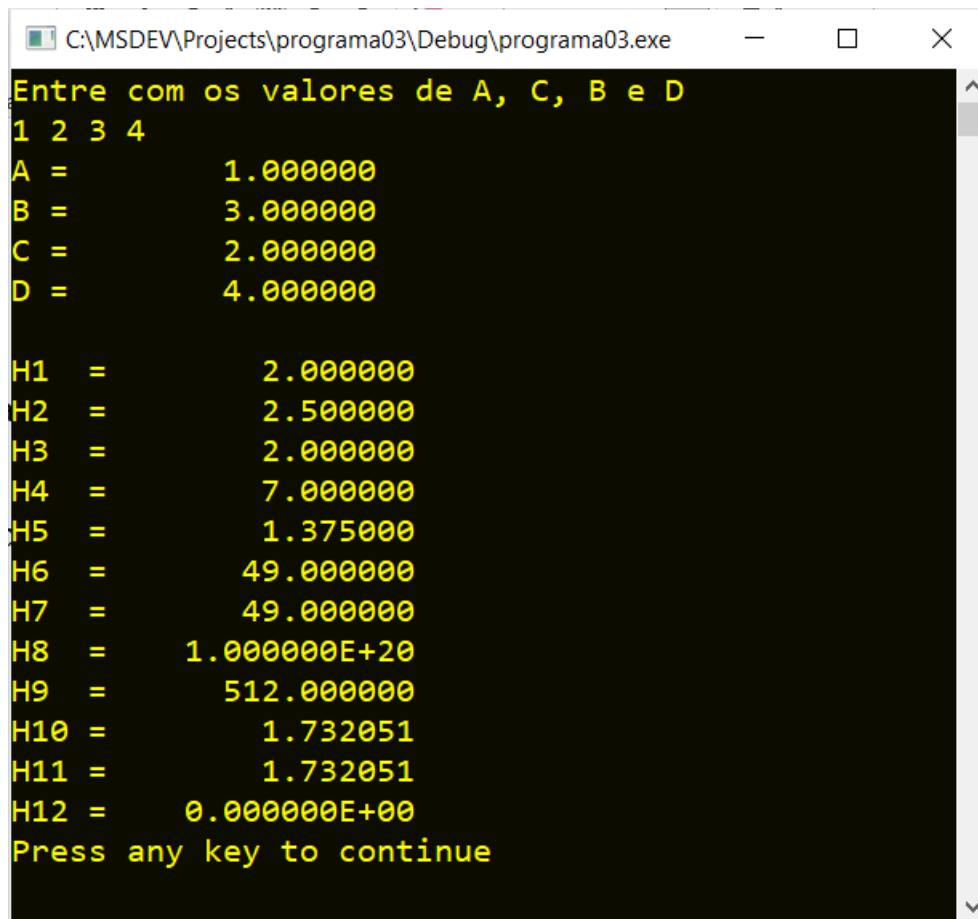
```

5) Comentários sobre o programa:

- Nas duas primeiras linhas é usado o comando REAL para definir as variáveis do tipo real do programa.
Em um mesmo programa, pode-se usar um ou mais comandos REAL ou INTEGER.
- A linha **READ(*,*) A, C, B, D** do programa é usada para ler os valores de quatro variáveis. Um único comando READ pode ser usado para a leitura de diversos valores que são atribuídos a diversas variáveis,

que têm que estar separadas por vírgula. Os valores são atribuídos às variáveis na mesma sequência que estas aparecem no comando **READ**.

- c) O comando **WRITE(*,*)** irá gerar uma linha em branco no dispositivo de saída, para separar os dados dos resultados.
 - d) As expressões em FORTRAN para as variáveis H1 a H12 correspondem respectivamente às expressões algébricas dadas pelas Equações (3.1) a (3.9).
- 6) Executar **Build, Compile** para compilar o programa
- 7) Gerar o programa-executável fazendo **Build, Build**.
- 8) Ao se executar o programa, através de **Build, Execute**, surge uma janela, mostrada na Figura 3.7, dentro da qual tem-se:



```
C:\MSDEV\Projects\programa03\Debug\programa03.exe
Entre com os valores de A, C, B e D
1 2 3 4
A =      1.000000
B =      3.000000
C =      2.000000
D =      4.000000

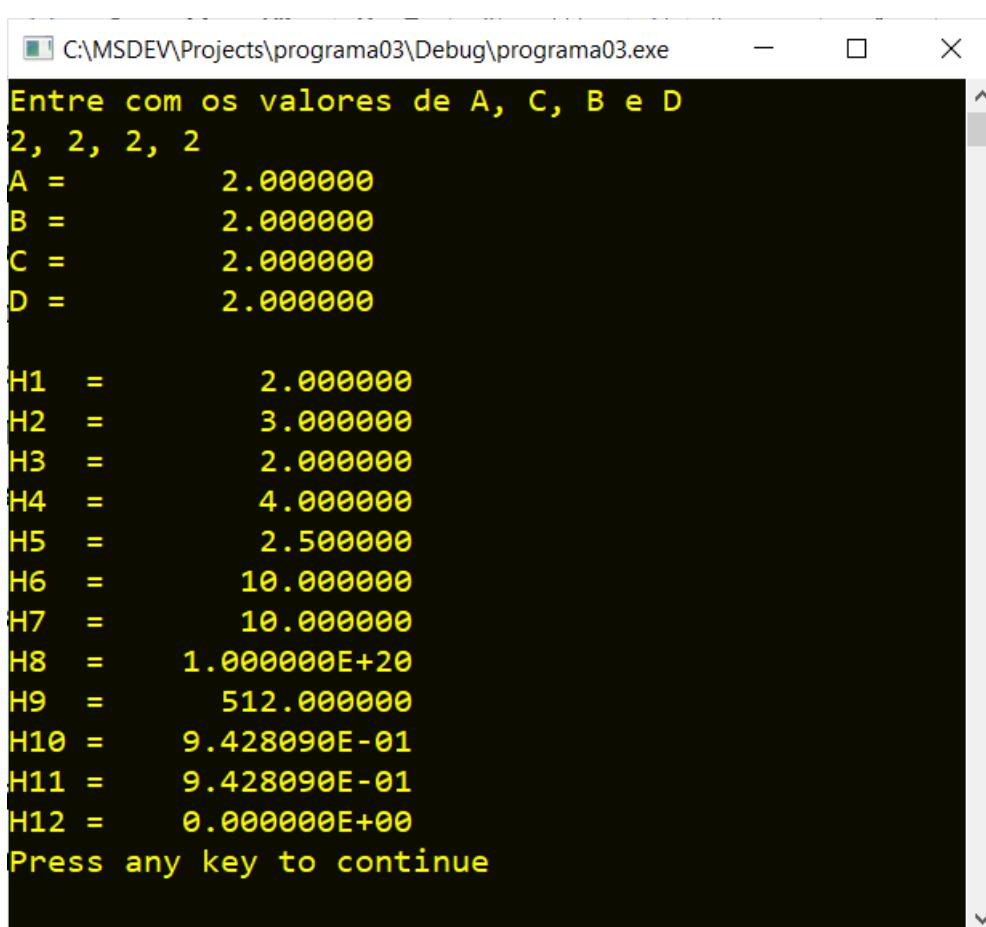
H1 =      2.000000
H2 =      2.500000
H3 =      2.000000
H4 =      7.000000
H5 =      1.375000
H6 =      49.000000
H7 =      49.000000
H8 =      1.000000E+20
H9 =      512.000000
H10 =     1.732051
H11 =     1.732051
H12 =     0.000000E+00
Press any key to continue
```

Figura 3.7 Resultado da execução do programa03f.f90 – exemplo 1.

- a) Na primeira linha, o comentário “Entre com os valores de A, C, B e D”, resultado do comando **WRITE(*,*) "Entre com os valores de A, C, B e D"** do programa.
- b) Na segunda linha, o programa pára e fica aguardando que sejam fornecidos os valores das variáveis A, C, B e D, resultado do comando **READ(*,*) A, C, B, D** do programa. Para que o programa continue sua execução é necessário **digitar quatro valores**. Cada valor digitado será atribuído respectivamente a uma das quatro variáveis, na mesma sequência. Há várias formas de se entrar com diversos valores para um único comando READ: digitando tudo na mesma linha, com cada valor separado por vírgula, separado

por um ou mais espaços em branco; digitando cada valor e, em seguida, clicando na tecla enter; ou misturas das três formas anteriores. A segunda forma foi usada no exemplo da Figura 3.7 para se entrar com os valores 1, 2, 3 e 4, respectivamente, para as variáveis A, C, B e D.

- c) Nas linhas seguintes, são confirmados os quatro dados e, em seguida, apresentados os resultados das variáveis H1 a H12.
- 9) **Analizar** cada resultado mostrado na Figura 3.7 comparando-o com o valor esperado obtido de um cálculo mental ou com o uso de uma calculadora. Para tanto, devem ser consideradas a escala de prioridades da Tabela 3.4 e as regras descritas na Tabela 3.5.
- 10) **Executar** novamente o programa com outros valores para as variáveis A, C, B e D: por exemplo, 2, 2, 2, 2. Analisar cada novo resultado comparando-o com o valor esperado obtido de um cálculo mental ou com o uso de uma calculadora. Os novos dados e resultados são mostrados na Figura 3.8.



```
C:\MSDEV\Projects\programa03\Debug\programa03.exe
Entre com os valores de A, C, B e D
2, 2, 2, 2
A =      2.000000
B =      2.000000
C =      2.000000
D =      2.000000

H1 =      2.000000
H2 =      3.000000
H3 =      2.000000
H4 =      4.000000
H5 =      2.500000
H6 =      10.000000
H7 =      10.000000
H8 =      1.000000E+20
H9 =      512.000000
H10 =     9.428090E-01
H11 =     9.428090E-01
H12 =     0.000000E+00
Press any key to continue
```

Figura 3.8 Resultado da execução do programa03f.f90 – exemplo 2.

3.6 EXERCÍCIOS

Exercício 3.1

- 1) Editar um programa-fonte em FORTRAN para executar o seguinte algoritmo (passos):
 - a) ler três números reais
 - b) calcular a média aritmética deles
 - c) escrever os valores lidos e o valor da média aritmética juntamente com comentários para identificá-los
- 2) Compilar o programa-fonte
- 3) Gerar o programa-executável
- 4) Executar o programa com os valores 1.0, 2.5 e -3.7. O resultado esperado é -6.666667E-2.
- 5) Executar o programa com os valores 10, -1E+3 e 2E-1. O resultado esperado é -3.299333E2.

Exercício 3.2

- 1) Ler os valores de I e J
- 2) Calcular os valores de K e M através de
$$K = \frac{I}{J} \quad M = \sqrt[4]{I+J}$$
- 3) Escrever o nome completo do aluno
- 4) Escrever os resultados de K e M junto com seus nomes
- 5) Executar o programa com $I = 2$ e $J = 3$.

Os resultados esperados são: $K = 0.6666667$ $M = 1.709976$

Exercício 3.3

- 1) Definir x, y, z como variáveis inteiras
- 2) Definir k como variável real
- 3) Ler os valores de x, y, z
- 4) Calcular
$$k = \frac{1}{2+x} + \sqrt[3]{z}$$
- 5) Escrever o resultado de k
- 6) Entrar com os dados $x = 5, y = 3$ e $z = 10$

O resultado esperado é $k = 2.297292$

Exercício 3.4

- 1) Editar um programa-fonte em FORTRAN para executar o seguinte algoritmo (passos):
 - a) ler o primeiro valor (real) de uma progressão aritmética (P.A.), denotado por A1
 - b) ler a diferença (número real) entre dois termos subsequentes da P.A., denotada por D
 - c) ler o número (inteiro) de termos da P.A., denotado por N
 - d) calcular o valor (real) do último termo da P.A., denotado por AN, com $AN = A1 + (N-1)*D$
 - e) calcular a soma de todos os termos da P.A., denotada por SN, com a seguinte expressão:
$$SN = (A1+AN)*N/2$$
 - f) escrever os três valores lidos e os dois calculados juntamente com comentários para identificá-los
- 2) Compilar o programa-fonte
- 3) Gerar o programa-executável
- 4) Executar o programa para $A1 = 1.3$, $D = 3.9$ e $N = 5$. Os resultados devem ser $AN = 16.9$ e $SN = 45.5$.

Exercício 3.5

Escrever as expressões em linguagem FORTRAN que correspondem às seguintes expressões algébricas, onde A , B e L são números reais e I e J são números inteiros:

$$a) A = \frac{\sqrt{B} + 5}{100}$$

$$b) A = \frac{1}{10} + \sqrt[3]{B}$$

$$c) L = \frac{1}{3-I} \sqrt{I^2 + \frac{1}{J^3}}$$

$$d) L = \frac{1}{1 + \frac{1}{I}} \left(I^2 + \frac{J}{I^3 - 2} \right)^{\frac{1}{3}}$$

Para verificar se a resposta de cada item está correta, basta implementar a expressão em FORTRAN em um programa e comparar seu resultado com o valor obtido através de uma calculadora para a expressão algébrica correspondente. Os dois resultados devem ser iguais. Isso também vale para o próximo exercício.

Exercício 3.6

Escrever as expressões algébricas que correspondem às seguintes expressões em linguagem FORTRAN:

- a) $A = ((B^{**}(1.0/2)) + 5) / 100$
- b) $A = (1.0/10) + (B^{**}(1/3.0))$
- c) $L = (1.0/(3.0-I)) * ((I^{**}2) + (1.0/(J^{**}3)))^{**}(1.0/2)$
- d) $L = (1.0 / (1.0 + (1.0/I))) * ((I^{**}2) + (J / (I^{**}3) - 2.0))^{**}(1.0/3)$