

Capítulo 13. Variáveis do tipo REAL DUPLA

30 Set 2025

OBJETIVOS DO CAPÍTULO

- Conceitos de: variáveis de precisão simples e dupla, variáveis do tipo real simples e dupla, mistura de variáveis do tipo inteiro com real simples e real dupla
- Funções matemáticas intrínsecas e formatos de edição para variáveis do tipo real dupla
- Comandos do FORTRAN: REAL, REAL*4, REAL(4), REAL*8, REAL(8), DOUBLE PRECISION e IMPLICIT NONE

13.1 programa13a.f90

1) Objetivos do programa:

- a) Definir uma variável do tipo real de dupla precisão usando o comando REAL*8
- b) Comparar a precisão de variáveis do tipo real simples e dupla

2) No Fortran, **criar um projeto** com o nome **programa13a**

3) No Fortran, **criar e inserir** no projeto o programa-fonte **programa13a.f90**

4) No Fortran, **copiar** exatamente o texto em vermelho mostrado na **Tabela 13.1**.

5) Comentários sobre o programa:

- a) No capítulo 3 foram abordadas as variáveis do tipo real. Elas são usadas 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. Para definir variáveis do tipo real nos programas, usa-se o comando REAL. As variáveis definidas com o comando REAL podem guardar na memória do computador valores no intervalo aproximado de 1.18e-38 a 3.40e+38, positivos e negativos, e a precisão é de 7 ou 8 algarismos significativos. Estas variáveis também podem ser chamadas de reais simples ou de precisão simples.
 - b) O novo comando REAL*8 é usado para definir variáveis do tipo real dupla ou de precisão dupla. As variáveis definidas com o comando REAL*8 podem guardar na memória do computador valores no intervalo aproximado de 2.23e-308 a 1.80e+308, positivos e negativos, e a precisão é de 15 ou 16 algarismos significativos. Em geral, variáveis do tipo REAL*8 são mais adequadas em cálculos de engenharia e computação científica. Porém, elas exigem o dobro da memória computacional em relação às variáveis reais simples.
 - c) A linha **REAL A, C** define as variáveis A e C como sendo do tipo real simples.
 - d) A linha **REAL*8 B, D, E, F** define as variáveis B, D, E e F como sendo do tipo real dupla.
 - e) É calculada a razão 2/3 para as variáveis C, E, E e F através de três formas.
- #### 6) Executar **Build, Compile** para compilar o programa

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

Tabela 13.1 Programa13a.f90.

```
REAL A, C
REAL*8 B, D, E, F

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

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

C = 2/3.0

D = 2/3.0

E = 2/3.0e0

F = 2/3.0d0

WRITE(*,*)
WRITE(*,*) "reais simples"
WRITE(*,*) "A = ", A
WRITE(*,*) "C = ", C

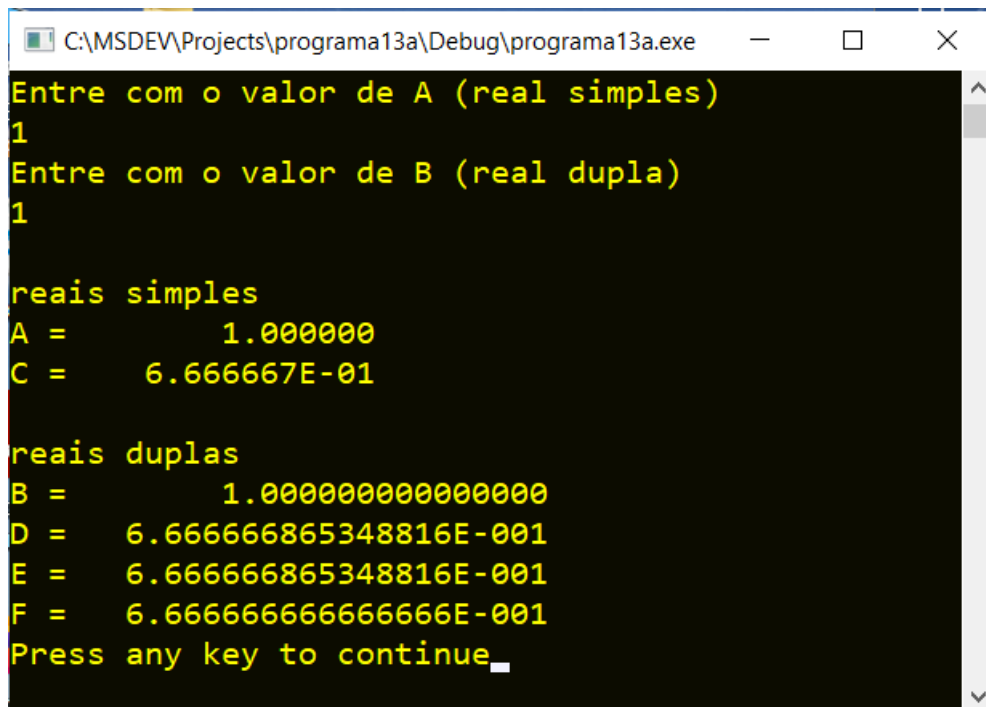
WRITE(*,*)
WRITE(*,*) "reais duplas"
WRITE(*,*) "B = ", B
WRITE(*,*) "D = ", D
WRITE(*,*) "E = ", E
WRITE(*,*) "F = ", F

END
```

8) Executar o programa através de **Build, Execute com o valor 1 para as duas variáveis**. O resultado deve ser o mostrado na Figura 13.1. Deve-se notar o seguinte:

- O número de algarismos significativos para as variáveis A e C, que são reais simples, é 7, tanto na notação decimal quanto na científica.
- O número de algarismos significativos para as variáveis B, D, E e F, que são reais duplas, é 16, tanto na notação decimal quanto na científica.
- A variável C, que é de precisão simples, teve o seu último dígito arredondado de 6 para 7, sabendo-se que $2/3 = 0.666666666666...$

- d) A variável F, que é de precisão dupla, não teve o seu último dígito arredondado de 6 para 7.
- e) As variáveis D e E, embora tenham sido definidas com precisão dupla, seus valores estão contaminados e, de fato, são de precisão simples. Isto ocorre devido à forma como a razão 2/3 foi calculada nelas. Pois, com variáveis de precisão dupla, sempre deve-se usar a letra “d” ou “D” em números reais para caracterizá-los como sendo de precisão dupla.



```
C:\MSDEV\Projects\programa13a\Debug\programa13a.exe
Entre com o valor de A (real simples)
1
Entre com o valor de B (real dupla)
1

reais simples
A = 1.000000
C = 6.66667E-01

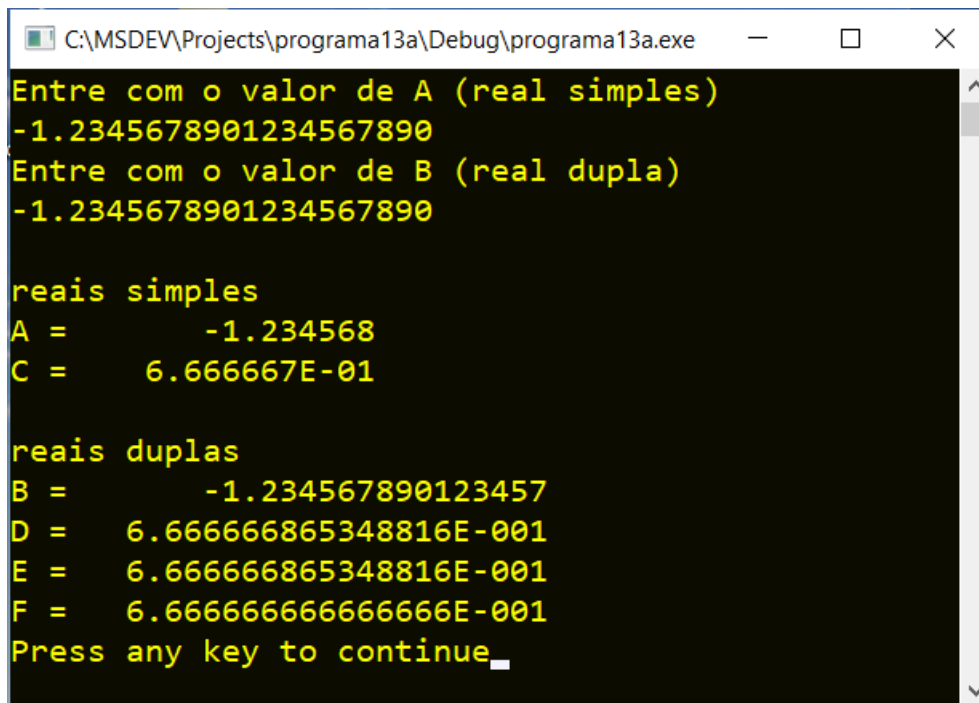
reais duplas
B = 1.0000000000000000
D = 6.66666865348816E-001
E = 6.66666865348816E-001
F = 6.66666666666666E-001
Press any key to continue
```

Figura 13.1 Resultado da execução do programa13a.f90 para A = B = 1.

- 9) Executar novamente o programa com o valor 123456789 para as duas variáveis e analisar os resultados.
- 10) Executar novamente o programa com o valor 0.001 para as duas variáveis e analisar os resultados.
- 11) Executar novamente o programa com o valor -1.2345678901234567890 para as duas variáveis e analisar os resultados, que são mostrados na Figura 13.2. Notar que o algarismo da última casa decimal, de cada precisão, é arredondado e os algarismos excedentes são desprezados.
- 12) Executar novamente o programa com outros dados e analisar os resultados.

13.2 programa13b.f90

- 1) Objetivos do programa:
- a) Usar diversas formas para definir variáveis reais simples e duplas
 - b) Realizar cálculos elementares envolvendo variáveis do tipo inteiro, real simples e real dupla
- 2) No Fortran, **criar um projeto** com o nome **programa13b**
- 3) No Fortran, **criar e inserir** no projeto o programa-fonte **programa13b.f90**
- 4) No Fortran, **copiar** exatamente o texto em vermelho mostrado na **Tabela 13.2**.



```
C:\MSDEV\Projects\programa13a\Debug\programa13a.exe
Entre com o valor de A (real simples)
-1.2345678901234567890
Entre com o valor de B (real dupla)
-1.2345678901234567890

reais simples
A = -1.234568
C = 6.666667E-01

reais duplas
B = -1.234567890123457
D = 6.666666865348816E-001
E = 6.666666865348816E-001
F = 6.666666666666666E-001
Press any key to continue
```

Figura 13.2 Resultado da execução do programa13a.f90 para $A = B = -1.2345678901234567890$.

5) Comentários sobre o programa:

- a) As variáveis do tipo real simples podem ser definidas através dos comandos REAL, REAL*4 e REAL(4). Exemplos são as definições das variáveis A, B e C.
- b) As variáveis do tipo real dupla podem ser definidas através dos comandos REAL*8, REAL(8) e DOUBLE PRECISION. Exemplos são as definições das variáveis E, F e G.
- c) Os cálculos das variáveis D, B e C são exatamente os mesmos mostrados no programa03d.f90 do capítulo 3, seção 3.2, para explicar as consequências de misturar variáveis do tipo inteiro e real simples em cálculos. Estas mesmas explicações valem para as variáveis reais dupla F e G.
- d) Quando se atribui uma variável real simples a uma real dupla, esta tem sua precisão deteriorada para real simples. Um exemplo é o cálculo da variável E. Isso deve sempre ser evitado. Se o objetivo é usar precisão dupla no programa, todas as variáveis e os cálculos devem envolver precisão dupla.
- e) As variáveis H e I exemplificam o efeito da soma ou subtração entre números com ordens (valores) muito diferentes; dependendo da precisão dos cálculos, o menor valor é desprezado.

6) Executar **Build, Compile** para compilar o programa

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

8) Executar o programa através de **Build, Execute com o valor 6.9 para a variável A**.

9) **Analisar os resultados mostrados na Figura 13.3.** Deve-se notar o seguinte:

- a) A variável D, por ser inteira, só consegue absorver a parte inteira da variável A.
- b) As variáveis B e F são nulas devido ao cálculo da razão $1/D$ que envolve dois números inteiros.
- c) A diferença de precisão no arredondamento do cálculo das variáveis C e G; destaca-se o uso de 1.0d0 no cálculo de G.

Tabela 13.2 Programa13b.f90.

```

INTEGER D

REAL      A, H
REAL*4    B
REAL(4)   C

REAL*8    E, I
REAL(8)   F
DOUBLE PRECISION G

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

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

E = A
F = 1 / D
G = 1.0d0 / D

H = 1 + 1.0e-10
I = 1 + 1.0d-10

WRITE(*,*)
WRITE(*,*) "inteiro"
WRITE(*,*) "D = ", D

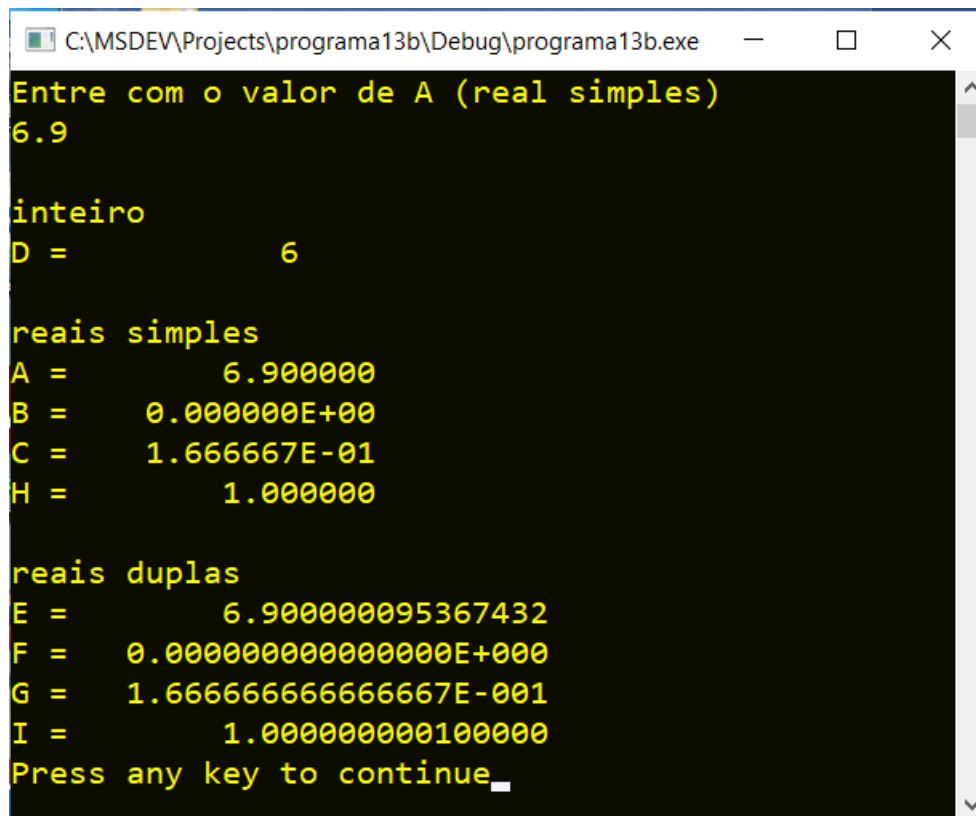
WRITE(*,*)
WRITE(*,*) "reais simples"
WRITE(*,*) "A = ", A
WRITE(*,*) "B = ", B
WRITE(*,*) "C = ", C
WRITE(*,*) "H = ", H

WRITE(*,*)
WRITE(*,*) "reais duplas"
WRITE(*,*) "E = ", E
WRITE(*,*) "F = ", F
WRITE(*,*) "G = ", G
WRITE(*,*) "I = ", I

END

```

- d) A deterioração da precisão da variável real dupla E causada ao se igualá-la à variável real simples A: todos os algarismos que aparecem na variável E após os zeros são lixo, também chamado de erro de arredondamento.
- e) O valor de H não conseguiu absorver $1.0\text{e-}10$ com 1, mas o valor de I, sim.
- 10) **Executar novamente o programa com outros valores para a variável A e analisar os resultados.**



```
C:\MSDEV\Projects\programa13b\Debug\programa13b.exe
Entre com o valor de A (real simples)
6.9

inteiro
D =          6

reais simples
A =          6.900000
B =          0.000000E+00
C =          1.666667E-01
H =          1.000000

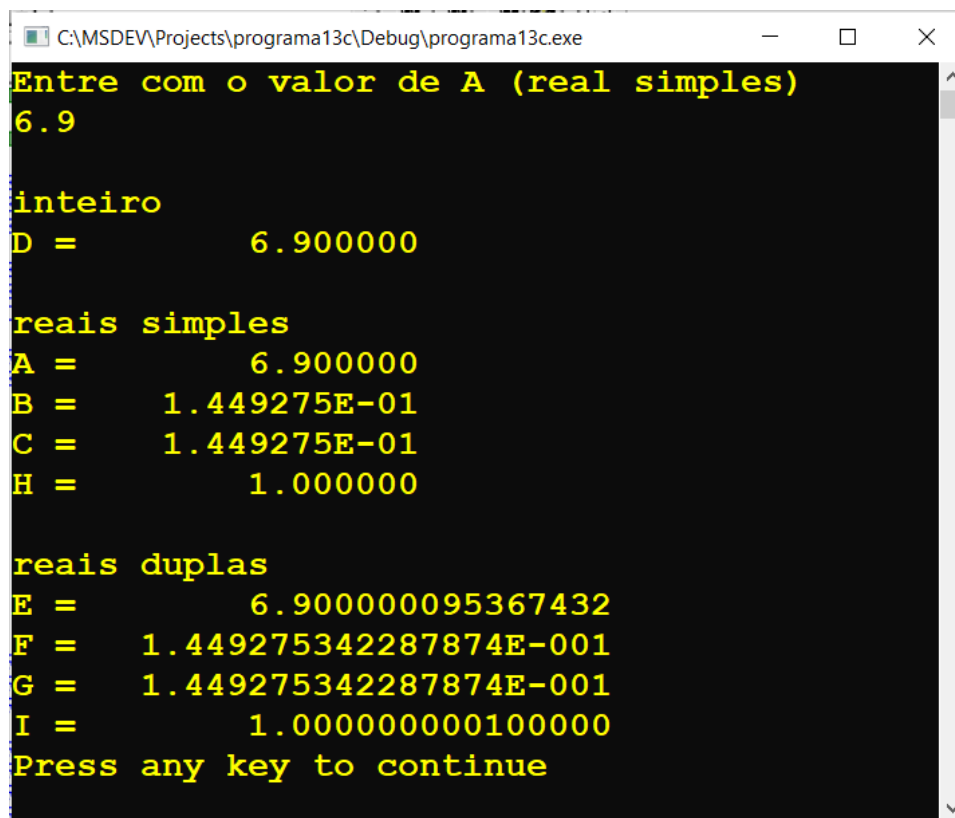
reais duplas
E =          6.900000095367432
F =          0.0000000000000000E+000
G =          1.666666666666667E-001
I =          1.000000000100000
Press any key to continue_
```

Figura 13.3 Resultado da execução do programa13b.f90 com A = 6.9.

13.3 programa13c.f90

- 1) Objetivos do programa:
 - a) Mostrar a importância de serem definidas todas as variáveis do programa
 - b) Utilizar o comando IMPLICIT NONE
- 2) No Fortran, **criar um projeto** com o nome **programa13c**
- 3) No Fortran, **criar e inserir** no projeto o programa-fonte **programa13c.f90**
- 4) No Fortran, **copiar** exatamente o texto em vermelho mostrado na **Tabela 13.2**, isto é, copiar o programa13b.f90 do projeto anterior.
- 5) Eliminar a primeira linha do programa, ou seja, **INTEGER D**.
- 6) Executar **Build, Compile** para compilar o programa. Mesmo a variável D não tendo sido definida, não ocorre erro de compilação. Quando uma variável não é definida, por default, assume-se que todas as variáveis iniciadas pelas letras I até N, em ordem alfabética, são do tipo inteiro, e as demais são do tipo real simples.
- 7) Gerar o programa-executável fazendo **Build, Build**.

- 8) Executar o programa através de **Build, Execute com o valor 6.9 para a variável A**. A princípio, os mesmos resultados mostrados na Figura 13.3 deveriam ser obtidos. Porém, pode-se notar na Figura 13.4 que isso não ocorre porque a variável D passou a ser uma variável real simples; com isso, foram alterados os valores das variáveis B, C, F e G.
- 9) **Incluir na primeira linha do programa13c.f90 o comando IMPLICIT NONE**. Ele deve ser colocado antes de qualquer declaração de tipo de variável. Recomenda-se fortemente sempre usar o comando IMPLICIT NONE porque ele obriga o programador a declarar explicitamente os tipos de todas as variáveis empregadas no programa. Assim, se houver a digitação errônea do nome de uma variável, o compilador apontará o erro.



```
C:\MSDEV\Projects\programa13c\Debug\programa13c.exe
Entre com o valor de A (real simples)
6.9

inteiro
D =          6.900000

reais simples
A =          6.900000
B =      1.449275E-01
C =      1.449275E-01
H =          1.000000

reais duplas
E =      6.900000095367432
F =      1.449275342287874E-001
G =      1.449275342287874E-001
I =          1.000000000100000
Press any key to continue
```

Figura 13.4 Resultado da execução do programa13c.f90 sem definir a variável inteira D.

- 10) Executar **Build, Compile** para compilar a nova versão do programa. Deve-se notar que agora ocorrerá erro de compilação, apontando que a variável D não foi declarada.
- 11) **Incluir a declaração da variável D como sendo do tipo inteiro.**
- 12) Executar **Build, Compile** para compilar o programa.
- 13) Gerar o programa-executável fazendo **Build, Build**.
- 14) Executar o programa através de **Build, Execute com o valor 6.9 para a variável A**. Agora os resultados são os mesmos mostrados na Figura 13.3, conforme esperado.

13.4 programa13d.f90

- 1) Objetivos do programa:
 - a) Usar funções matemáticas intrínsecas do FORTRAN para variáveis reais duplas
 - b) Usar formatos de edição para variáveis reais duplas
- 2) No Fortran, **criar um projeto** com o nome **programa13d**
- 3) No Fortran, **criar e inserir** no projeto o programa-fonte **programa13d.f90**
- 4) No Fortran, **copiar** exatamente o texto em vermelho mostrado na **Tabela 13.3**.

Tabela 13.3 Programa13d.f90.

```
USE PORTLIB
IMPLICIT NONE
REAL    A4, B4, C4, Pi4
REAL*8  A8, B8, C8, Pi8
INTEGER VER

WRITE(*,*) "Entre com o valor de A8 (real dupla)"
READ(*,*) A8

OPEN(9,file="saida13d.txt")

A4 = A8
WRITE(9,*) "A4 = A8 =", A4

B4 = ABS(A4)
B8 = DABS(A8)
WRITE(9,*) "B4 = Módulo com ABS(A4) =", B4
WRITE(9,*) "B8 = Módulo com DABS(A8) =", B8

C4 = LOG10(B4)
C8 = DLOG10(B8)
WRITE(9,*) "C4 = Logaritmo decimal com LOG10(B4) =", C4
WRITE(9,*) "C8 = Logaritmo decimal com DLOG10(B8) =", C8

Pi4 = ACOS(-1.0e0)
Pi8 = DACOS(-1.0d0)
WRITE(9,*) "Pi com ACOS(-1.0e0) = ", Pi4
WRITE(9,*) "Pi com DACOS(-1.0d0) = ", Pi8

WRITE(9,*) A8, "= A8 sem formato"
WRITE(9,21) A8, A8, A8, A8, A8
WRITE(9,11) A8, B8, C8
WRITE(9,12) A8, B8, C8
```



```
CLOSE(9)
```

```
VER = SYSTEM("Notepad saida13d.txt")
```

```
11 FORMAT( 5X, "A8 =", 1PE15.3, 1/,      &  
          5X, "B8 =", 1PE15.3, 1/,      &  
          5X, "C8 =", 1PE15.3, 1/       )
```

```
12 FORMAT ( 3(1PE10.1), "= variáveis A8, B8 e C8 no formato 3(1PE10.1)" )
```

```
21 FORMAT ("A8 no formato F8.2      =", F8.2,      1/, &  
          "A8 no formato F5.0      =", F5.0,      1/, &  
          "A8 no formato E15.3E1   =", E15.3E1,   1/, &  
          "A8 no formato 1PE15.2   =", 1PE15.2,   1/, &  
          "A8 no formato 1PE15.2E4 =", 1PE15.2E4, 1/  )
```

```
END
```

5) Comentários sobre o programa:

- a) Os tipos de funções matemáticas intrínsecas do FORTRAN são os mesmos para variáveis reais simples ou duplas. Mas os nomes são distintos para preservar a precisão das variáveis reais duplas; em geral, acrescenta-se a letra D à frente do comando usado com variáveis do tipo real simples. Por exemplo: a função ABS(X), usada para calcular o módulo da variável X do tipo real simples, passa a ser DABS quando X é do tipo real dupla.
- b) As Tabelas 13.4 e 13.5 mostram algumas funções matemáticas intrínsecas do FORTRAN para variáveis do tipo real dupla. Estas tabelas são as correspondentes às Tabelas 5.5 e 5.6 para variáveis do tipo real simples.

Tabela 13.4 Algumas funções matemáticas intrínsecas do FORTRAN com argumentos e resultados sendo variáveis do tipo real dupla.

Função	Comando	Observação
Módulo ou valor absoluto	Y = DABS(X)	
Raiz quadrada	Y = DSQRT(X)	
Exponencial	Y = DEXP(X)	
Logaritmo natural	Y = DLOG(X)	
Logaritmo decimal	Y = DLOG10(X)	
Parte inteira de número real	Y = IDINT(X)	Y = parte inteira de X
Resto da divisão	Y = DMOD(X,Z)	Y = X – INT(X/Z)*Z; se Y = 0, a divisão é exata; se Y ≠ 0, há resto diferente de 0 na divisão
Sinal	Y = DSIGN(X,Z)	Y = valor absoluto de X vezes o sinal de Z
Mínimo	Y = DMIN1(X,Z,W)	Y = valor mínimo entre X, Z e W; o argumento pode conter 2 ou mais variáveis
Máximo	Y = DMAX1(X,Z,W)	Y = valor máximo entre X, Z e W; o argumento pode conter 2 ou mais variáveis

Tabela 13.5 Algumas funções trigonométricas do FORTRAN com argumentos e resultados sendo variáveis do tipo real dupla.

Função	Comando	Observação
Seno	$Y = \text{DSIN}(X)$	X em radianos
Seno	$Y = \text{DSIND}(X)$	X em graus
Cosseno	$Y = \text{DCOSD}(X)$	X em graus
Tangente	$Y = \text{DTAND}(X)$	X em graus
Arco-seno	$Y = \text{DASIND}(X)$	Y em graus
Arco-cosseno	$Y = \text{DACOSD}(X)$	Y em graus
Arco-tangente	$Y = \text{DATAND}(X)$	Y em graus
Seno hiperbólico	$Y = \text{DSINH}(X)$	X em radianos
Cosseno hiperbólico	$Y = \text{DCOSH}(X)$	X em radianos
Tangente hiperbólica	$Y = \text{DTANH}(X)$	X em radianos

- c) A lista completa de funções matemáticas intrínsecas do FORTRAN pode ser vista no manual do Fortran. Para acessá-lo, dentro da subjanela do lado esquerdo, deve-se executar: clicar uma vez sobre o símbolo **?InfoView**; e acessar as opções **Reference, Procedures, Numeric Procedures** e as opções **Reference, Procedures, Trigonometric, Exponential, Root, and Logathmic Procedures**.
- d) Os tipos de formatos de edição usados com variáveis do tipo real simples também podem ser usados com variáveis do tipo real dupla. Estes formatos foram vistos no capítulo 6, seção 6.3. Um outro formato que pode ser conveniente é IPEX.YEZ, onde a letra E indica expoente e Z o número de algarismos para o expoente. Por exemplo, no programa13d.f90 são usados os formatos E15.3E1 e 1PE15.2E4.
- e) Por default, quando não se usa o comando FORMAT e não se especifica o número de algarismos do expoente, as variáveis reais duplas são apresentadas com três algarismos, enquanto que as variáveis reais simples, com dois.
- f) Porém, por default, quando se usa o comando FORMAT mas não se especifica o número de algarismos do expoente, tanto as variáveis reais duplas quanto as reais simples são apresentadas com dois algarismos no expoente.
- g) Nos argumentos de funções intrínsecas para variáveis do tipo real dupla deve-se definir os números usando-se a letra D (ou d) em vez de E (ou e) (das variáveis reais simples) para representar o expoente. Em cálculos ou expressões que envolvam números reais, também deve-se fazer isso. Se estas recomendações não forem seguidas ocorrerá erro de compilação ou perda de precisão nos cálculos. Por exemplo, no programa13d.f90 usa-se $-1.0\text{e}0$ como argumento da função real simples ACOS para calcular o valor de π , e $-1.0\text{d}0$ como argumento da função real dupla DACOS.
- h) A escala de prioridades entre os operadores matemáticos e as regras adotadas nas operações matemáticas, descritas nas Tabelas 3.4 e 3.5 também se aplicam às variáveis do tipo real dupla, onde pertinente.
- 6) Executar **Build, Compile** para compilar o programa
- 7) Gerar o programa-executável fazendo **Build, Build**.
- 8) Executar o programa através de **Build, Execute com o valor -1.234567890 para a variável A8**.

- 9) Analisar os resultados mostrados na Figura 13.5.
- 10) Executar novamente o programa com outros valores para a variável A8 e analisar os resultados; por exemplo, $A8 = 1e+3$, $1.0d-3$, $1.2e-30$. Notar que para este último número, o valor não é apresentado para o formato E15.3E1, sendo apresentados apenas asteriscos que indicam o uso de um formato inadequado.

```

saida13d - Bloco de Notas
Arquivo  Editar  Formatar  Exibir  Ajuda
A4 = A8 =          -1.234568
B4 = Módulo com  ABS(A4) =          1.234568
B8 = Módulo com DABS(A8) =          1.2345678900000000
C4 = Logaritmo decimal com LOG10(B4) =    9.151497E-02
C8 = Logaritmo decimal com DLOG10(B8) =   9.151497716927041E-002
Pi com ACOS(-1.0e0) =          3.141593
Pi com DACOS(-1.0d0) =          3.141592653589793
      -1.2345678900000000= A8 sem formato
A8 no formato F8.2      =    -1.23
A8 no formato F5.0      =    -1.
A8 no formato E15.3E1   =          -.123E+1
A8 no formato 1PE15.2   =          -1.23E+00
A8 no formato 1PE15.2E4 =          -1.23E+0000

      A8 =          -1.235E+00
      B8 =          1.235E+00
      C8 =          9.151E-02

-1.2E+00   1.2E+00   9.2E-02= variáveis A8, B8 e C8 no formato 3(1PE10.1)

```

Figura 13.5 Resultado da execução do programa13d.f90 no arquivo saida13d.txt com $A8 = -1.234567890$.

13.5 EXERCÍCIOS

Exercício 13.1

Para testar as operações matemáticas básicas, refazer o programa03e.f90, do capítulo 3, seção 3.3, usando variáveis do tipo real dupla em vez de real simples.

Exercício 13.2

Para testar o uso da escala de prioridades entre os operadores matemáticos e as regras adotadas nas operações matemáticas, descritas nas Tabelas 3.4 e 3.5, refazer o programa03f.f90, do capítulo 3, seção 3.5, usando variáveis do tipo real dupla em vez de real simples, escrever os resultados H1 a H12 em arquivo tipo texto, e abrir automaticamente o arquivo de resultados com o aplicativo Bloco de Notas.

Comparar os resultados com aqueles da Figura 3.7.

Exercício 13.3

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

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

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

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

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

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 13.4

- 1) Executar o programa13d.f90 com $A8 = 1e50$, que significa 1×10^{50} . Deve-se notar que o programa terá problema durante a execução. Por que isso ocorre?
- 2) Implementar uma nova versão do programa13d.f90 com a eliminação de tudo relacionado às variáveis reais de precisão simples.
 - a) Gerar o executável desta nova versão e executá-lo com $A8 = 1e50$. Deve-se notar que agora o programa é executado sem ocorrer o problema do item 1. Por que isso acontece? Deve-se notar também que em alguns formatos aparecem asteriscos.
 - b) Executar novamente o programa com $A8 = 2.2e-308$.
 - c) Executar novamente o programa com $A8 = 2.0e+308$. O que acontece? Por quê?
 - d) Executar novamente o programa com outros valores para $A8$.