

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

Para inicializar as atividades deste capítulo, deve-se acessar o programa Fortran, no Windows através de: **Start, Programs, Fortran PowerStation 4.0, Microsoft Developer Studio**

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, seguindo o [procedimento-padrão](#), **criar um projeto** com o nome **programa13a**
- 3) No Fortran, seguindo o [procedimento-padrão](#), **criar e inserir** no projeto o programa-fonte **programa13a.f90**
- 4) Dentro do espaço de edição do Fortran, na subjanela maior, **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 uma variável do tipo real num programa usa-se o comando REAL. As variáveis definidas com o comando REAL podem guardar na memória do computador valores no intervalo aproximado de 1.17e-38 a 3.40e+38 e a precisão é de 6 casas decimais. 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 2.22e-308 a 1.79e+308 e a precisão é de 15 casas decimais.

Em geral, estas variáveis são mais adequadas em computação científica. Entretanto elas exigem o dobro da memória computacional em relação às variáveis reais simples.

- c) A linha **REAL A** define a variável A como sendo do tipo real simples.
- d) A linha **REAL*8 B** define a variável B como sendo do tipo real dupla.

Tabela 13.1 Programa13a.f90.

```
REAL A
REAL*8 B

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

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

WRITE(*,*) "Valor de A = ", A
WRITE(*,*) "Valor de B = ", B

END
```

- 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 para as duas variáveis**. O resultado deve ser o mostrado na Figura 13.1. Deve-se notar que o número de casas decimais da variável A (real simples) é 6, e da variável B (real dupla) é 15.

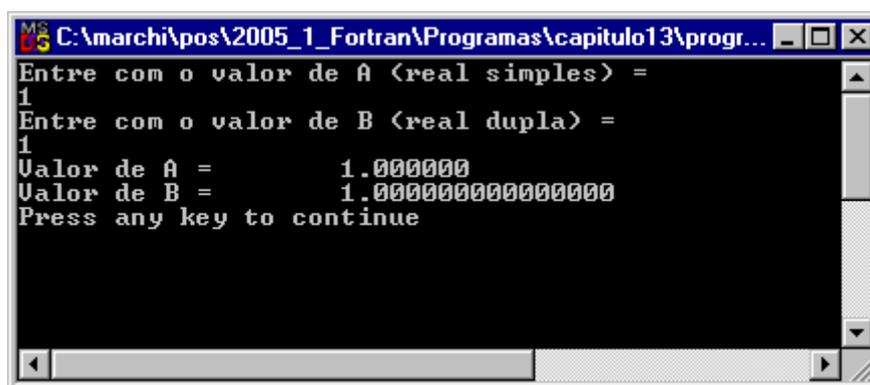


Figura 13.1 Resultado da execução do programa13a.f90 com A e B = 1.

- 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. Notar que o algarismo da última casa decimal é arredondado e os algarismos excedentes são desprezados.
- 12) Executar novamente o programa com outros dados e analisar os resultados.
- 13) No Fortran, para fechar o projeto atual, executar **File, Close Workspace**

13.2 programa13b.f90

- 1) Objetivos do programa:
 - a) Usar diversas formas para definir variáveis reais simples e dupla
 - b) Realizar cálculos elementares envolvendo variáveis do tipo inteiro, real simples e real dupla
- 2) No Fortran, seguindo o **procedimento-padrão**, **criar um projeto** com o nome **programa13b**
- 3) No Fortran, seguindo o **procedimento-padrão**, **criar e inserir** no projeto o programa-fonte **programa13b.f90**
- 4) Dentro do espaço de edição do Fortran, na subjanela maior, **copiar** exatamente o texto em vermelho mostrado na **Tabela 13.2**.
- 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 conseqüências de misturar variáveis do tipo inteiro e real simples num cálculo. 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 num programa, todas as variáveis e os cálculos devem envolver precisão dupla, exceto em alguns tipos de cálculo com variáveis inteiras.
- 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 13.2.** Deve-se notar a diferença de precisão no arredondamento do cálculo das variáveis C e G. Também deve-se notar a deterioração da precisão da

variável real dupla E causada ao se igualar ela à variável real simples A: todos os algarismos que aparecem na variável E após os zeros são lixo, também chamado de erro de arredondamento.

Tabela 13.2 Programa13b.f90.

```
INTEGER D

REAL    A
REAL*4  B
REAL(4) C

REAL*8  E
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 = 1.0 / D

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

WRITE(*,*) "Valor de B (real simples) = ", B
WRITE(*,*) "Valor de C (real simples) = ", C

WRITE(*,*) "Valor de E (real dupla)   = ", E
WRITE(*,*) "Valor de F (real dupla)   = ", F
WRITE(*,*) "Valor de G (real dupla)   = ", G

END
```

- 10) Executar novamente o programa com outros valores para a variável A e analisar os resultados.
- 11) No Fortran, para fechar o projeto atual, executar **File, Close Workspace**

```

C:\marchi\pos\2005_1_Fortran\Programas\capitulo13\programa13b\Debu...
Entre com o valor de A (real simples) =
6.9
Valor de D (inteiro) =          6
Valor de B (real simples) =    0.000000E+00
Valor de C (real simples) =    1.666667E-01
Valor de E (real dupla)  =    6.900000095367432
Valor de F (real dupla)  =    0.0000000000000000E+000
Valor de G (real dupla)  =    1.6666666666666667E-001
Press any key to continue_

```

Figura 13.2 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 de um programa
 - b) Utilizar o comando IMPLICIT NONE
- 2) No Fortran, seguindo o [procedimento-padrão](#), **criar um projeto** com o nome **programa13c**
- 3) No Fortran, seguindo o [procedimento-padrão](#), **criar e inserir** no projeto o programa-fonte **programa13c.f90**
- 4) Dentro do espaço de edição do Fortran, na subjanela maior, **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**. Em princípio, os mesmos resultados mostrados na Figura 13.2 deveriam ser obtidos. Porém, pode-se notar na Figura 13.3 que isso não ocorre, ressaltando-se: 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.

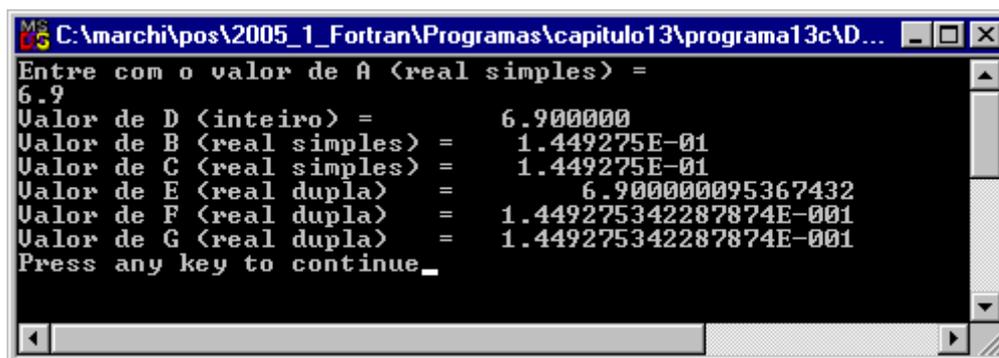


Figura 13.3 Resultado da execução do programa13b.f90 sem definir a variável inteira D.

- 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**, conforme é mostrado na Tabela 13.3.
- 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 13.2, conforme esperado.
- 15) No Fortran, para fechar o projeto atual, executar **File, Close Workspace**

Tabela 13.3 Programa13c.f90.

```
IMPLICIT NONE

INTEGER D

REAL    A
REAL*4  B
REAL(4) C

REAL*8  E
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 = 1.0 / D

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

WRITE(*,*) "Valor de B (real simples) = ", B
WRITE(*,*) "Valor de C (real simples) = ", C

WRITE(*,*) "Valor de E (real dupla) = ", E
WRITE(*,*) "Valor de F (real dupla) = ", F
WRITE(*,*) "Valor de G (real dupla) = ", G

END

```

13.4 programa13d.f90

- 1) Objetivos do programa:
 - a) Usar funções matemáticas intrínsecas do FORTRAN com variáveis reais do tipo dupla
 - b) Usar formatos de edição com variáveis reais do tipo dupla
- 2) No Fortran, seguindo o [procedimento-padrão](#), **criar um projeto** com o nome **programa13d**
- 3) No Fortran, seguindo o [procedimento-padrão](#), **criar e inserir** no projeto o programa-fonte **programa13d.f90**
- 4) Dentro do espaço de edição do Fortran, na subjanela maior, **copiar** exatamente o texto em vermelho mostrado na **Tabela 13.4**.

Tabela 13.4 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 E15.3     =", E15.3,     1/, &
          "A8 no formato 1PE15.2   =", 1PE15.2,   1/, &
          "A8 no formato 1PE15.2E3 =", 1PE15.2E3, 1/  )

END

```

5) Comentários sobre o programa:

- a) Os tipos de funções matemáticas intrínsecas do FORTRAN são os mesmos para variáveis reais simples ou dupla. Mas os nomes são distintos para preservar a precisão das variáveis reais dupla; 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 13.5 e 13.6 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 13.5 Algumas funções matemáticas 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: $Y = e^x$	Y = DEXP(X)	
Logaritmo natural	Y = DLOG(X)	
Logaritmo decimal	Y = DLOG10(X)	
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

- 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 1PEX.YEZ, onde a terminação EZ indica expoente e Z o número de algarismos para o expoente. Por exemplo, no programa13d.f90 usa-se o formato 1PE15.2E3.
- e) 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 $-1.0e0$ como argumento da função real simples ACOS para calcular o valor de π , e $-1.0d0$ como argumento da função real dupla DACOS.

- f) 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.

Tabela 13.6 Algumas funções trigonométricas do FORTRAN com argumentos e resultados sendo variáveis do tipo real dupla.

Função	Comando	Observação
Seno	$Y = DSIN(X)$	X em radianos
Cosseno	$Y = DCOS(X)$	X em radianos
Tangente	$Y = DTAN(X)$	X em radianos
Arco-seno	$Y = DASIN(X)$	Y em radianos
Arco-cosseno	$Y = DACOS(X)$	Y em radianos
Arco-tangente	$Y = DATAN(X)$	Y em radianos
Seno hiperbólico	$Y = DSINH(X)$	
Cosseno hiperbólico	$Y = DCOSH(X)$	
Tangente hiperbólica	$Y = DTANH(X)$	

- 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 13.4.**
- 10) **Executar novamente o programa com outros valores para a variável A8 e analisar os resultados; por exemplo, $A8 = 1e+3, 1.0d-3, 1.2e-30$.**
- 11) Encerrar a sessão seguindo o [procedimento-padrão](#).

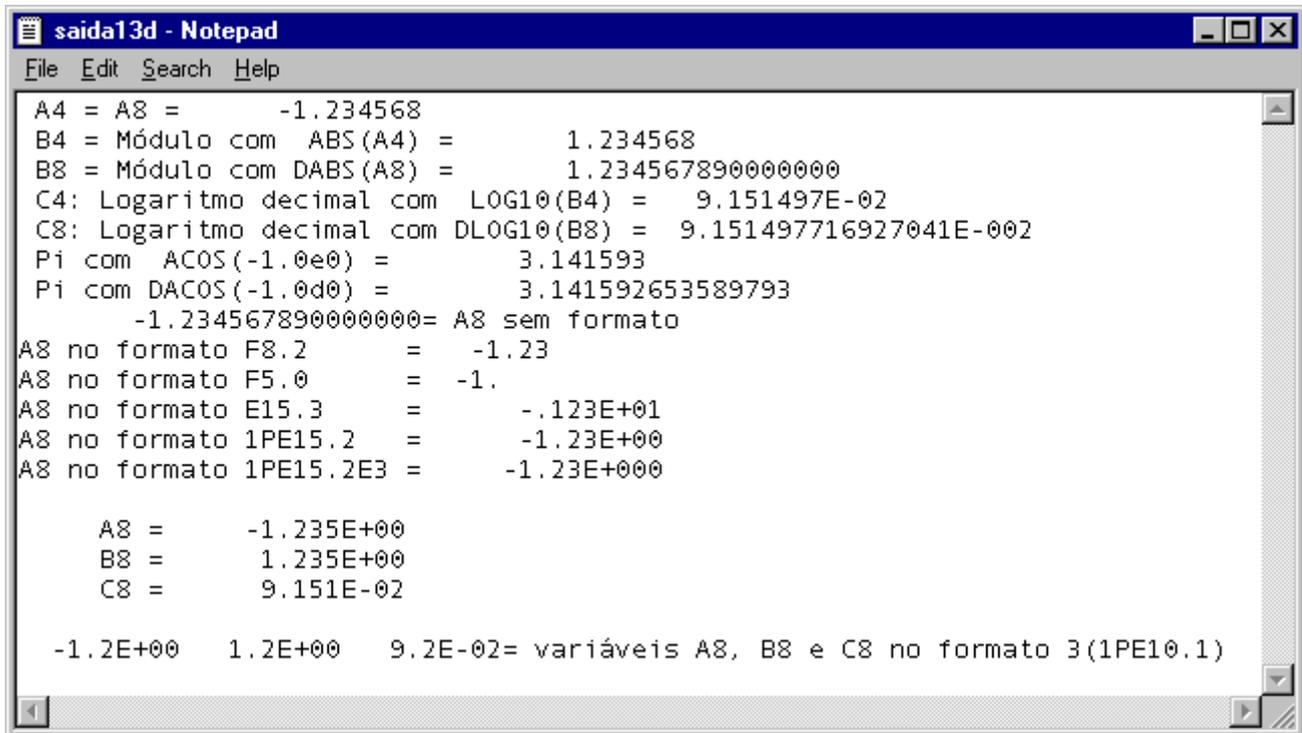
13.5 EXERCÍCIOS

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, onde pertinente, variáveis do tipo real dupla em vez de real simples.



```
saida13d - Notepad
File Edit Search Help
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 E15.3 = -.123E+01
A8 no formato 1PE15.2 = -1.23E+00
A8 no formato 1PE15.2E3 = -1.23E+000

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

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:

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

Exercício 13.4

- 1) Executar o programa13d.f90 com $A8 = 1.2e+308$. 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 $A8 = 1.2e+308$. Deve-se notar que agora o programa é executado sem nenhum problema. Por que isso ocorre? Observar o que acontece com a representação dos resultados através dos formatos usados, principalmente com relação aos expoentes.
 - 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 $A8$.