

Capítulo 2. VARIÁVEIS DO TIPO INTEIRO

OBJETIVOS DO CAPÍTULO

- Conceitos de: variáveis do tipo inteiro, atribuição, avisos e erros de compilação, erros de execução, comentários dentro do programa-fonte
- Operadores matemáticos básicos
- Comandos do FORTRAN: INTEGER e READ

2.1 programa02.f90, versão A

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

- 1) No Windows, seguindo o procedimento apresentado na seção 1.5 do capítulo 1: **Start, Programs, Fortran PowerStation 4.0, Microsoft Developer Studio**
- 2) No Fortran, seguindo o procedimento apresentado na seção 1.6 do capítulo 1, criar um projeto do tipo Console Application com o nome **programa02** no diretório **Z:\SERVER1\Alunos2004_1\login**, onde login deve ser substituído pelo user name do usuário, isto é, a conta particular na rede Windows do DEMEC/UFPR.
- 3) No Fortran, seguindo o procedimento apresentado na seção 1.7 do capítulo 1, criar e inserir no projeto o programa-fonte **programa02.f90**
- 4) Conforme é mostrado na Figura 2.1, dentro do espaço de edição do Fortran, na subjanela maior, copiar exatamente o texto mostrado abaixo em vermelho.

```
INTEGER A  
WRITE (*, *) "A"  
WRITE (*, *) A  
END
```

- 5) Comentários sobre o programa:
 - a) No capítulo 1 foram usados os comandos WRITE e END da linguagem FORTRAN. No programa02.f90, há um novo comando: INTEGER. Ele é usado para definir variáveis do tipo inteiro, isto é, variáveis que podem guardar ou armazenar na memória do computador números inteiros, positivos ou negativos, como 2, 5, 0, 54367 ou -3.
 - b) A linha **INTEGER A** define a variável A como sendo do tipo inteiro. Este comando reserva um espaço na memória do computador, utilizando o nome ou rótulo A para armazenar um valor inteiro.

- c) A linha `WRITE (*, *) "A"` escreve o comentário que está entre aspas; no caso a letra A.
 - d) A linha `WRITE (*, *) A` escreve o valor da variável A que está armazenado na memória do computador.
 - e) A linha `END` encerra o programa.
- 6) Ao se compilar o programa, executando **Build, Compile**, o resultado deve ser o mostrado na Figura 2.1. Deve-se notar na subjanela inferior um aviso (warning) mencionando que o valor da variável A não foi definido.
- 7) Gerar o programa-executável fazendo **Build, Build**.

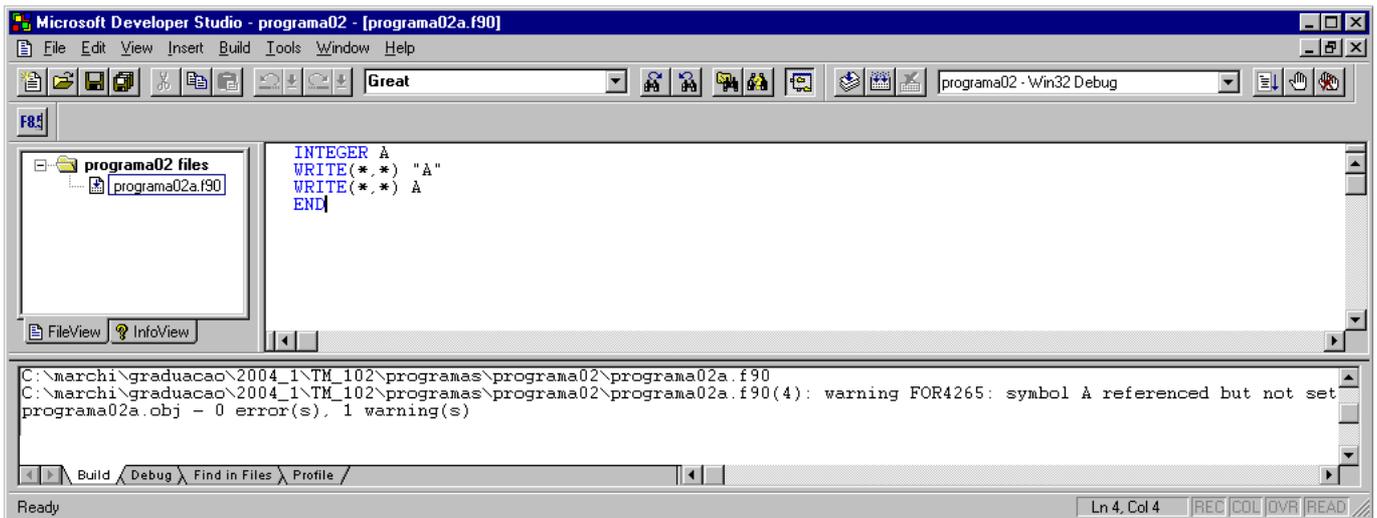


Figura 2.1 Resultado da compilação do programa02.f90, versão A.

- 8) Ao se executar o programa, através de **Build, Execute**, surge uma janela, mostrada na Figura 2.2, dentro da qual tem-se:
- a) Na primeira linha, a letra A, resultado do comando `WRITE (*, *) "A"` do programa.
 - b) Na segunda linha, o valor zero, resultado do comando `WRITE (*, *) A` do programa. Isso ocorre porque não foi definido um valor para a variável A, conforme o próprio Fortran informou durante a compilação do programa. Portanto, sempre é necessário definir o valor de cada variável do programa, caso contrário, por default, assume-se valor nulo.
 - c) E na terceira 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.
- 9) Deve-se perceber a diferença que existe entre os comandos `WRITE (*, *) "A"` e `WRITE (*, *) A` do programa. No primeiro, A é um comentário. No segundo, A é uma variável utilizada para armazenar um valor inteiro na memória do computador.

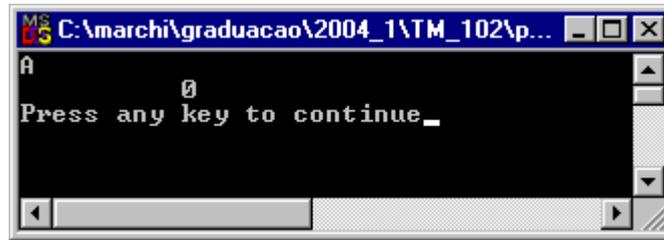


Figura 2.2 Resultado da execução do programa02.f90, versão A.

2.2 programa02.f90, versão B

- 1) Dentro do Fortran, editar exatamente o texto mostrado abaixo em vermelho.

```
INTEGER A
A = 3
WRITE (*,*) "A"
WRITE (*,*) A
END
```

- 2) Comentários sobre o programa: a única diferença entre a versão anterior (A) e a atual (B) do programa02.f90 é a inclusão da segunda linha, ou seja, **A = 3**. O sinal de igualdade dentro de um programa escrito em linguagem FORTRAN é utilizado para atribuir o valor que está do lado direito à variável do lado esquerdo. Portanto, neste caso, o valor 3 é atribuído à variável A. Em outras palavras, o valor 3 é armazenado num espaço da memória do computador que é identificado pelo nome ou rótulo A, o nome da variável. Este valor utilizado (3) é apenas um exemplo; ele pode ser qualquer número inteiro.
- 3) Nesta versão do programa, ao se executar **Build, Compile**, não haverá aviso (warning) porque, neste caso, o valor da variável A está definido.
- 4) Gerar o programa-executável fazendo **Build, Build**.
- 5) Ao se executar o programa, através de **Build, Execute**, surge uma janela, mostrada na Figura 2.3, dentro da qual tem-se:
 - a) Na primeira linha, a letra A, resultado do comando **WRITE (*,*) "A"** do programa.
 - b) Na segunda linha, o valor 3, resultado do comando **WRITE (*,*) A** do programa e do comando anterior, **A = 3**.
 - c) E na terceira linha, a frase Press any key to continue.
- 6) Deve-se perceber que o programa é executado, linha por linha, da primeira (**INTEGER A**) até a última (**END**).
- 7) Um exemplo de erro de compilação é apresentado na Figura 2.4. Ele ocorre devido à eliminação do segundo asterisco da terceira linha do programa. Erros de compilação ocorrem quando os comandos

do FORTRAN são utilizados de forma incorreta. Na subjanela inferior do Fortran, geralmente haverá um comentário indicando cada erro (error). Logo após o nome do programa-fonte compilado, entre parênteses, é indicado o número da linha do programa-fonte aonde deve estar o erro. No exemplo da Figura 2.4, o compilador do Fortran mostra o seguinte comentário: C:\marchi\programa02\programa02b.f90(3): error FOR3852: syntax error detected between , and). Portanto, este comentário indica que na linha 3 há um erro de sintaxe (erro que resulta do uso incorreto de um comando, no caso o WRITE) entre a vírgula e o sinal de fechamento de parênteses.

- 8) As linhas do programa-fonte são numeradas do topo para baixo, e as colunas, da esquerda para a direita. Na extremidade inferior da janela principal do Fortran, do lado direito, sempre são indicados a linha (Ln) e a coluna (Col) aonde o cursor se encontra dentro do programa-fonte.

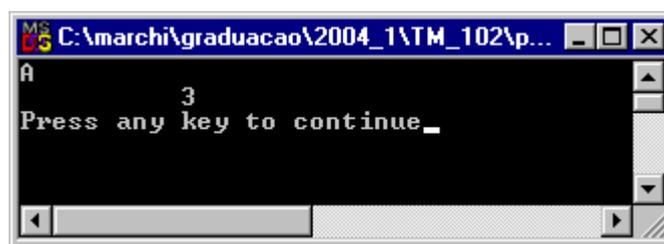


Figura 2.3 Resultado da execução do programa02.f90, versão B.

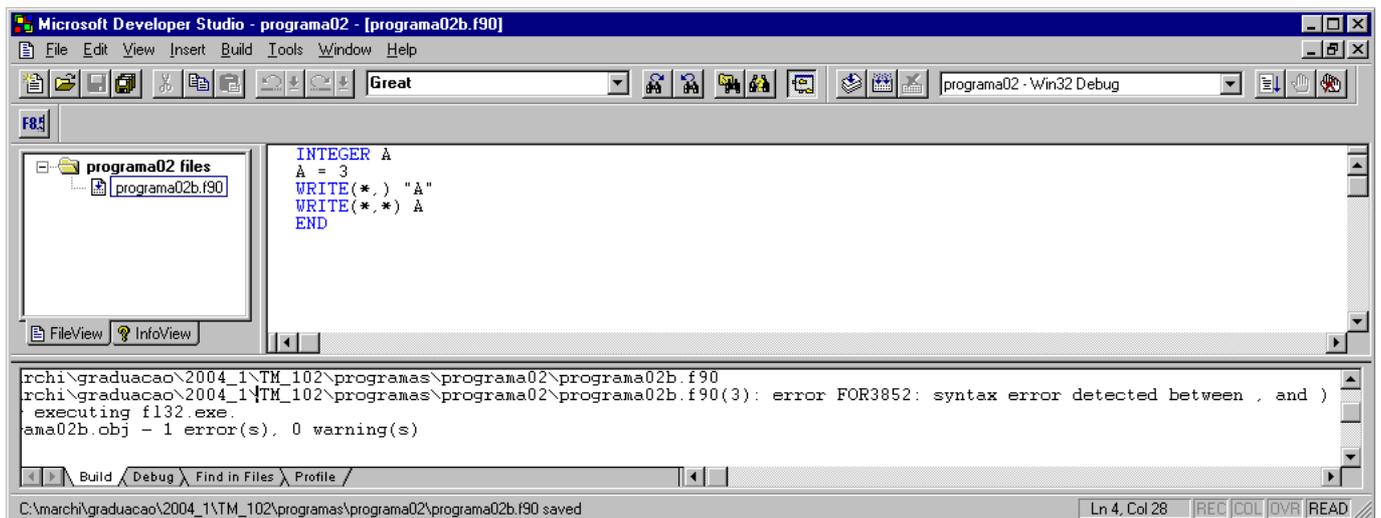


Figura 2.4 Exemplo de erro de compilação.

2.3 programa02.f90, versão C

- 1) Dentro do Fortran, editar exatamente o texto mostrado abaixo em vermelho.

```

INTEGER A
A = 3
WRITE(*,*) "Valor de A = ", A

```

END

- 2) Comentários sobre o programa: a diferença básica entre a versão anterior (B) e a atual (C) do programa02.f90 é a junção dos dois comandos WRITE num só, na terceira linha do programa, isto é, `WRITE(*,*) "Valor de A = ", A`. Esta forma do comando WRITE é usada quando se quer escrever na mesma linha diversos elementos. No caso, são apenas dois elementos, ou seja, o comentário Valor de A = e a variável A. Os elementos devem ser separados por vírgula.
- 3) Executar **Build, Compile** para compilar o programa.
- 4) Gerar o programa-executável fazendo **Build, Build**.
- 5) Ao se executar o programa, através de **Build, Execute**, surge uma janela, mostrada na Figura 2.5, dentro da qual tem-se:
 - a) Na primeira linha, como resultado do comando `WRITE(*,*) "Valor de A = ", A` do programa, o comentário Valor de A = e, na mesma linha, o valor da variável A, cujo valor atribuído dentro do programa foi 3.
 - b) E na segunda linha, a frase Press any key to continue.

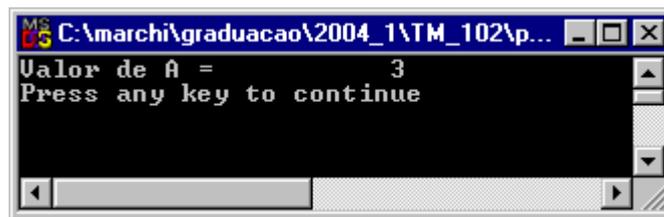


Figura 2.5 Resultado da execução do programa02.f90, versão C.

2.4 programa02.f90, versão D

- 1) Dentro do Fortran, editar exatamente o texto mostrado abaixo em vermelho.

```
INTEGER A
A = 4
A = 3
WRITE(*,*) "Valor de A = ", A
END
```

- 2) Comentários sobre o programa: a única diferença entre a versão anterior (C) e a atual (D) do programa02.f90 é a inclusão da linha `A = 4`, que atribui o valor 4 à variável A.
- 3) Executar **Build, Compile** para compilar o programa.
- 4) Gerar o programa-executável fazendo **Build, Build**.

- 5) Ao se executar o programa, através de **Build, Execute**, o resultado é o mesmo mostrado na Figura 2.5. Isso ocorre porque o programa é executado, linha por linha, da primeira (**INTEGER A**) até a última (**END**). Assim, embora tenha sido atribuído o valor 4 à variável A na segunda linha do programa, na linha seguinte atribui-se o valor 3 à mesma variável, e, na quarta linha do programa, escreve-se o valor de A. O valor escrito é 3 porque é o último valor que foi armazenado na memória do computador. A denominação variável é usada justamente porque seu valor pode ser alterado ao longo da execução do programa.

2.5 programa02.f90, versão E

- 1) Dentro do Fortran, editar exatamente o texto mostrado abaixo em vermelho, incluindo a linha em branco.

```
! Programa02.f90
INTEGER A

A = 3 ! atribui o valor 3 à variável A
WRITE(*,*) "Valor de A = ", A
END
```

- 2) Comentários sobre o programa: em cada linha do programa-fonte, tudo que estiver à direita do símbolo ! (exclamação) não é executado pelo programa. São apenas comentários usados para esclarecer o que faz cada parte do programa. Isso é chamado de documentação interna. Dentro do editor do Fortran, todos os comentários ficam na cor verde, como é mostrado na Figura 2.6. Um comentário pode envolver uma linha inteira do programa, como na primeira da versão E, ou apenas uma parte, como na quarta linha do programa. Linhas em branco dentro do programa-fonte também não são executadas. Elas equivalem a um comentário em branco. Um exemplo é a terceira linha do programa02.f90, versão E, na Figura 2.6.
- 3) Executar **Build, Compile** para compilar o programa.
- 4) Gerar o programa-executável fazendo **Build, Build**.
- 5) Ao se executar o programa, através de **Build, Execute**, o resultado é o mesmo mostrado na Figura 2.5. Isso ocorre porque as diferenças entre a versão C e a atual (E) do programa02.f90 são apenas os comentários e uma linha em branco, que não são executados pelo programa.
- 6) Para maior clareza e facilidade de compreensão do programa-fonte, recomenda-se que dentro dele sejam usados comentários e linhas em branco.

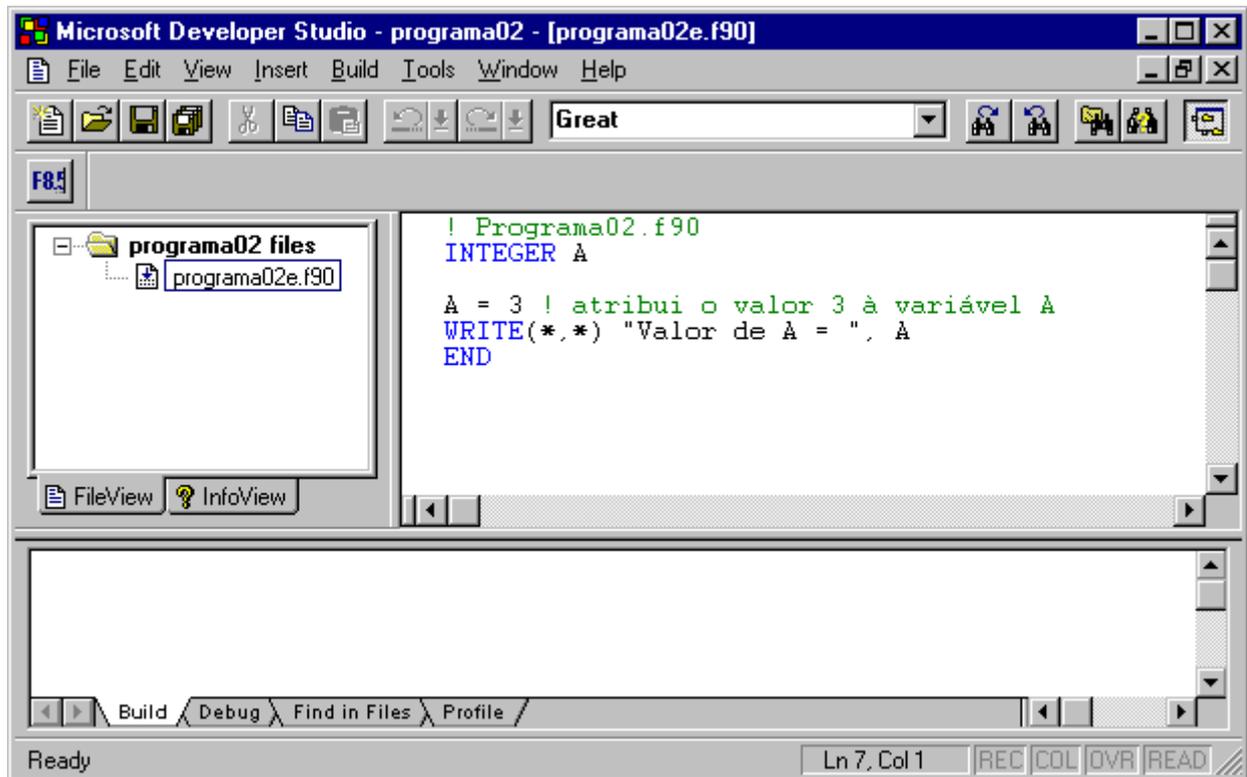


Figura 2.6 Programa02.f90, versão E.

2.6 programa03.f90, versão A

Para inicializar as atividades com o próximo programa, deve-se executar:

- 1) No Fortran, para fechar o projeto atual: **File, Close Workspace**
- 2) Na janela Microsoft Developer Studio, clicar sobre o botão **Yes**
- 3) No Fortran, seguindo o procedimento apresentado na seção 1.6 do capítulo 1, criar um projeto do tipo Console Application com o nome **programa03** no diretório **Z:\SERVER1\Alunos2004_1\login**, onde login deve ser substituído pelo user name do usuário, isto é, a conta particular na rede Windows do DEMEC/UFPR.
- 4) No Fortran, seguindo o procedimento apresentado na seção 1.7 do capítulo 1, criar e inserir no projeto o programa-fonte **programa03.f90**
- 5) Dentro do Fortran, editar exatamente o texto mostrado abaixo em vermelho, incluindo as linhas em branco.

INTEGER A, B, C, D, E, F, G

A = -6

B = 2

C = A + B

D = B - A

```

E = A * B
F = A / B
G = A ** B

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

END

```

6) Comentários sobre o programa:

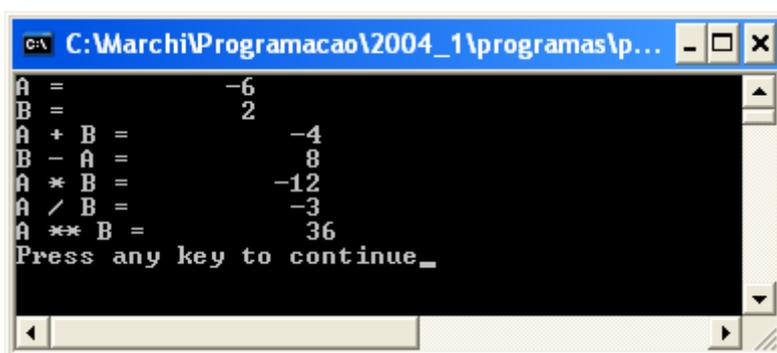
- a) A linha `INTEGER A, B, C, D, E, F, G` define as variáveis A, B, C, D, E, F e G como sendo do tipo inteiro. Este comando reserva espaço na memória do computador para diversas variáveis com apenas um comando `INTEGER`. Entretanto, as variáveis devem estar separadas por vírgula.
- b) As linhas `A = -6` e `B = 2` atribuem os valores inteiros -6 e 2 às variáveis A e B. Estes valores são apenas exemplos; eles podem ser quaisquer números inteiros.
- c) As variáveis C, D, E, F e G são calculadas em função dos valores das variáveis A e B, usando os cinco operadores matemáticos básicos definidos na Tabela 2.1, conforme explicado a seguir.

Tabela 2.1 Operadores matemáticos básicos em FORTRAN.

Símbolo do operador matemático em FORTRAN	Nome do símbolo	Operação matemática correspondente
+	Sinal mais	Adição
-	Sinal menos	Subtração
*	Asterisco	Multiplicação
/	Barra	Divisão
**	Duplo asterisco	Potenciação

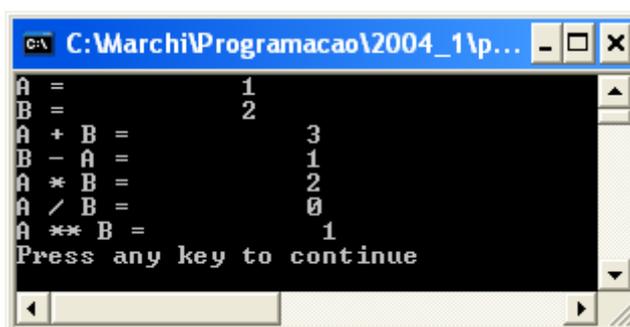
- d) A linha `C = A + B` adiciona os valores das variáveis A e B e atribui o resultado à variável C.
- e) A linha `D = B - A` subtrai o valor da variável A do valor da variável B e atribui o resultado à variável D.
- f) A linha `E = A * B` multiplica os valores das variáveis A e B e atribui o resultado à variável E.
- g) A linha `F = A / B` divide o valor da variável A pelo valor da variável B e atribui o resultado à variável F.

- h) A linha `G = A ** B` eleva o valor da variável A ao valor da variável B e atribui o resultado à variável G.
- 7) Executar **Build, Compile** para compilar o programa.
 - 8) Gerar o programa-executável fazendo **Build, Build**.
 - 9) Ao se executar o programa, através de **Build, Execute**, surge uma janela, mostrada na Figura 2.7, dentro da qual tem-se: os valores das variáveis A e B; e os valores resultantes das cinco operações matemáticas básicas da Tabela 2.1 efetuadas com as variáveis A e B. **Analisar** cada resultado, comparando-o com o valor esperado obtido de um cálculo mental.
 - 10) **Dentro do programa-fonte, fazer A = 1 e B = 2. Compilar e gerar o executável** do programa com esta alteração. **Executar o programa** cujo resultado deve ser aquele mostrado na Figura 2.8. **Analisar** cada resultado, comparando-o com o valor esperado obtido de um cálculo mental. O resultado da divisão pode parecer incorreto mas não é. Isso se deve ao seguinte: 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 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 divisão mostrado na Figura 2.8. Este tema será explorado com mais detalhes no Capítulo 3.



```
C:\Marchi\Programacao\2004_1\programas\p...
A = -6
B = 2
A + B = -4
B - A = 8
A * B = -12
A / B = -3
A ** B = 36
Press any key to continue_
```

Figura 2.7 Resultado da execução do programa03.f90, versão A, com A = -6 e B = 2.



```
C:\Marchi\Programacao\2004_1\p...
A = 1
B = 2
A + B = 3
B - A = 1
A * B = 2
A / B = 0
A ** B = 1
Press any key to continue
```

Figura 2.8 Resultado da execução do programa03.f90, versão A, com A = 1 e B = 2.

11) Atribuir valores às variáveis dentro do próprio programa-fonte não é recomendável. Pois, para alterar os valores, é necessário ter o programa-fonte, além de recompilá-lo e gerar o programa-executável a cada vez. O procedimento mais indicado é utilizar o comando READ, apresentado na próxima seção.

2.7 programa03.f90, versão B

1) Dentro do Fortran, alterar o programa03.f90, versão A, para que fique exatamente igual ao texto mostrado abaixo em vermelho, incluindo as linhas em branco.

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

END
```

2) Comentários sobre o programa:

- a) A única diferença entre a versão anterior (A) e a atual (B) do programa03.f90 está no início do programa. É a inclusão de 4 linhas novas no lugar de se atribuir valores às variáveis A e B.
- b) Até aqui, os comandos da linguagem FORTRAN que foram usados são: WRITE, END e INTEGER. Na versão B do programa03.f90, há um novo comando: READ. Ele é usado para atribuir (fornecer) valores às variáveis durante a execução de um programa. Isto é, o comando READ é empregado para LER os dados de um programa.

- c) A linha `WRITE(*,*) "Entre com o valor de A"` escreve o comentário que está entre aspas.
- d) A linha `READ(*,*) A` lê um valor digitado dentro da janela DOS, aberta durante a execução do programa, e o atribui à variável A.
- 3) Executar **Build, Compile** para compilar o programa.
- 4) Gerar o programa-executável fazendo **Build, Build**.
- 5) Ao se executar o programa, através de **Build, Execute**, surge uma janela do tipo DOS 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 -6 para a variável A** e, em seguida, **clique 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 2 para a variável B** e, em seguida, **clique na tecla enter**.
- e) Em seguida, são apresentados os mesmos resultados da versão A do programa03.f90, conforme a Figura 2.9.

```

C:\Marchi\Programacao\2004_1\programas\p...
Entre com o valor de A
-6
Entre com o valor de B
2
A = -6
B = 2
A + B = -4
B - A = 8
A * B = -12
A / B = -3
A ** B = 36
Press any key to continue_

```

Figura 2.9 Resultado da execução do programa03.f90, versão B, com A = -6 e B = 2.

- 6) **Executar novamente o programa, entrando com outros valores para A e B.** Em seguida, **analisar** cada resultado, comparando-o com o valor esperado obtido de um cálculo mental.
- 7) **Executar novamente o programa, com A = 6 e B = 0.** Nenhum resultado é apresentado porque o programa não consegue dividir 6 por 0. Isso gera um erro que interrompe a execução normal do programa. Ao se implementar um programa, deve-se prepará-lo para que seja evitado qualquer

divisão por zero. Isso pode ser feito através de comentários que informem ao usuário do programa, na janela da execução do programa ou no manual do programa, por exemplo, quais as variáveis que não podem ter valor nulo. Outra forma mais efetiva de evitar divisão por zero será vista em capítulo futuro.

2.8 FIM DA SESSÃO

Para concluir o uso do Fortran e do Windows, deve-se executar o seguinte:

- 1) Para encerrar as atividades com um projeto, basta executar no menu do Fortran: **File, Close Workspace**
- 2) Na janela Microsoft Developer Studio, clicar sobre o botão **Yes**
- 3) Para fechar o aplicativo Fortran, basta fazer o seguinte em seu menu: **File, Exit**
- 4) Para fechar o Windows, executar: **Start, Shut Down...**
- 5) Na janela Shut Down Windows, escolher a opção **Close all programs and log on as a different user?**. Clicar sobre o botão **Yes**

2.9 EXERCÍCIOS

Exercício 2.1

Executar novamente o programa03.f90, versão B, com $A = 2$ e $B = -1$. Em seguida, analisar cada resultado, comparando-o com o valor esperado obtido de um cálculo mental, especialmente o caso da potenciação. No Fortran, para abrir um projeto já existente, como o programa03, basta executar **File, Open Workspace**. Em seguida, **indicar o diretório do projeto e selecionar o arquivo que tem o nome do projeto e extensão mdp**, no caso programa03.mdp

Exercício 2.2

Executar novamente o programa03.f90, versão B, usando números reais, isto é, $A = 1.5$ e $B = 0.4$. Números reais são representados com o sinal de ponto para separar a parte inteira da decimal.

Exercício 2.3

- 1) Editar um programa-fonte em FORTRAN para executar o seguinte algoritmo (passos):
 - a) ler três números inteiros
 - 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, 2 e 3. Em seguida, analisar o resultado da média fornecido pelo programa comparando-o com o valor esperado obtido por um cálculo mental.
- 5) Repetir o item 4 para os valores 1, 1 e 2.

Exercício 2.4

- 1) Editar um programa-fonte em FORTRAN para executar o seguinte algoritmo (passos):
 - a) ler o primeiro valor (inteiro) de uma progressão aritmética (P.A.), denotado por A_1
 - b) ler a diferença (número inteiro) entre dois termos subseqüentes da P.A., denotada por D
 - c) ler o número (inteiro) de termos da P.A., denotado por N
 - d) calcular o último termo da P.A., denotado por A_N
 - e) calcular a soma de todos os termos da P.A., denotado por S_N
 - 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 $A_1 = 1$, $D = 3$ e $N = 5$. Os resultados devem ser $A_N = 13$ e $S_N = 35$.