

Capítulo 6. FORMATOS DE EDIÇÃO

OBJETIVOS DO CAPÍTULO

- Conceitos e comandos de formatos de edição para variáveis do tipo inteiro, caracter e real
- Regras de edição de programas-fonte em FORTRAN
- Seqüência obrigatória dos comandos dentro de programas-fonte em FORTRAN
- Comandos do FORTRAN: WRITE(número,número) e FORMAT

6.1 FORMATOS DE EDIÇÃO PARA VARIÁVEIS DO TIPO CHARACTER: programa6a.f90

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

- 1) Para acessar o programa Fortran, no Windows: **Start, Programs, Fortran PowerStation 4.0, Microsoft Developer Studio**
- 2) No Fortran, seguindo o [procedimento-padrão](#), **criar um projeto** com o nome **programa06**
- 3) No Fortran, seguindo o [procedimento-padrão](#), **criar e inserir** no projeto o programa-fonte **programa6a.f90**
- 4) Dentro do espaço de edição do Fortran, na subjanela maior, **copiar** exatamente o texto em vermelho mostrado na **Tabela 6.1**.

Tabela 6.1 Programa6a.f90.

```
USE PORTLIB

CHARACTER(10) A, B
CHARACTER(20) C
INTEGER VER

WRITE(*,*) "Variaveis do tipo caracter"
WRITE(*,*) "Entre com os conteudos de A e B"
READ(*,*) A, B

C = A // B

OPEN(9, file="saida6a.txt")

WRITE(9,*) A, "= A sem formato"
```

```

WRITE (9,21) A
21 FORMAT ( A2, "= A no formato A2" )

WRITE (9,22) A
22 FORMAT ( A5, "= A no formato A5" )

WRITE (9,23) A
23 FORMAT ( A10, "= A no formato A10" )

WRITE (9,24) A
24 FORMAT("10X", 10X, "Conteúdo de A =", A10, 1/)

WRITE (9,25) A
25 FORMAT("T10", T10, "Conteúdo de A =", A10, 1/)

WRITE (9,10) A, B, C
10 FORMAT(5X, "A =", A10, 1/, 5X, "B =", A10, 1/, 5X, "C =", A20, 2/)

WRITE (9,11) A, B, C
11 FORMAT( 5X, "A =", A10, 1/,      &
          5X, "B =", A10, 1/,      &
          5X, "C =", A20, 2/      )

WRITE (9,12) A, B
12 FORMAT ( 2(A10), "= variáveis A e B no formato 2(A10)" )

CLOSE (9)

VER = SYSTEM("Notepad saida6a.txt")

END

```

5) Comentários sobre o programa:

- a) O objetivo do programa6a.f90 é explicar alguns formatos de edição utilizados com variáveis do tipo caracter.
- b) No capítulo 5, além de um asterisco, o comando WRITE foi utilizado com um número para identificar o “local” (janela ou arquivo) onde seria escrito algo, isto é, ele foi usado na forma WRITE(número1,*). Um exemplo é a linha `WRITE (9,*) A, "= A sem formato"` do programa. Agora, no programa6a.f90, o comando WRITE é utilizado com dois números, ou seja, na forma WRITE(número1,número2), por exemplo, a linha `WRITE (9,21) A` do programa. O número2 está associado a um novo comando do FORTRAN chamado de FORMAT.

- c) O comando FORMAT é usado para especificar o formato com que comentários e conteúdos (strings ou números) de variáveis são escritos numa janela ou num arquivo. Também é usado na leitura de dados.
- d) A sintaxe, isto é, a forma com que se deve usar o comando FORMAT é: número2 FORMAT (formatos). O número2 deve ser o mesmo utilizado no comando WRITE. Dentro de parênteses, deve-se especificar os formatos de edição que serão usados, separados por vírgula. Um exemplo é a linha `21 FORMAT (A2, "= A no formato A2")` do programa, que especifica o formato A2 para escrever a variável A, do tipo character, chamada na linha `WRITE(9,21) A` do programa.
- e) Os comandos FORMAT podem ser incluídos em qualquer lugar dentro do programa-fonte, desde que seja abaixo de comandos USE e acima do comando END. Eles podem ser aglomerados num único lugar ou cada um pode estar abaixo ou acima do seu respectivo comando WRITE. Não é necessário seguir qualquer ordenamento para os números usados nos comandos FORMAT, nem para suas posições relativas dentro do programa-fonte. E não existe qualquer relação entre os números usados em comandos OPEN e os números usados nos comandos FORMAT. Eles até podem ser iguais.
- f) Cada tipo de variável (character, inteiro ou real) tem identificadores especiais para escrever seu conteúdo na forma desejada. Além disso, existem identificadores que facilitam a escrita de comentários e da forma ou disposição estética com que se escreve algo.
- g) O identificador de variáveis do tipo character é a letra A seguida de um número que especifica a quantidade máxima de caracteres para escrever cada variável. Dentro deste espaço, o conteúdo de cada variável do tipo character é escrito da esquerda para a direita. Se o conteúdo da variável ocupa mais espaço do que o reservado para escrevê-la, a parte excedente não é escrita, é desconsiderada. Um exemplo de uso deste identificador é A2, citado no item d, acima. Neste caso, reservou-se duas colunas ou caracteres para escrever uma variável do tipo character.
- h) Dentro de um comando FORMAT, isto é, entre parênteses, pode-se escrever um ou mais comentários, cada um devendo ficar entre aspas, como na linha `22 FORMAT (A5, "= A no formato A5")` do programa.
- i) O identificador X é usado para deixar espaços em branco entre dois formatos de edição. Ele é usado na forma NX, onde N é um número que representa a quantidade de espaços em branco. Um exemplo é a linha `24 FORMAT("10X", 10X, "Conteúdo de A =", A10, 1/)` do programa, onde usou-se o formato 10X.
- j) Nesta mesma linha do programa há o formato de edição N/, onde N é um número que representa a quantidade de linhas que o cursor descera quando encontrar este formato. Neste caso, uma linha, devido ao formato 1/.

- k) O identificador T, de tabulação, é usado para se escrever algo a partir de uma coluna específica, onde a coluna 1 fica na margem esquerda. Ele é usado na forma TN, onde N é o número da coluna a partir da qual se escreverá algo que vem a seguir dentro de um comando FORMAT. Um exemplo é a linha `25 FORMAT("T10", T10, "Conteúdo de A =", A10, 1/)` do programa, onde usou-se o formato T10.
- l) Pode-se usar um único comando WRITE, com seu respectivo comando FORMAT, para se escrever diversas variáveis. Neste caso, cada variável será escrita com o formato que estiver na mesma seqüência dos comandos WRITE e FORMAT. Por exemplo, na linha `WRITE(9,10) A, B, C` do programa, escreve-se as variáveis A, B e C com o formato número 10, que corresponde à linha `10 FORMAT(5X, "A =", A10, 1/, 5X, "B =", A10, 1/, 5X, "C =", A20, 2/)` do programa. Portanto, a primeira variável (A) será escrita com o formato A10, que é o primeiro formato de edição de variáveis que aparece no comando FORMAT. E as variáveis B e C serão escritas com os formatos A10 e A20, respectivamente, que são o segundo e o terceiro formatos de variáveis no comando FORMAT.
- m) O comando FORMAT identificado pelo número 10 no programa tem exatamente a mesma função que o comando FORMAT identificado pelo número 11. A única diferença é na edição do programa-fonte. No 10 FORMAT, todos os formatos de edição usam apenas uma linha do programa-fonte. Já no 11 FORMAT, são usadas três linhas. Neste caso, quando um mesmo comando precisa ou se quer empregar mais de uma linha por motivos estéticos, usa-se o símbolo &, ao final de cada linha do programa-fonte, para indicar que o comando continua na linha seguinte. Isso vale para qualquer comando do FORTRAN.
- n) O formato M(AN) é equivalente a M vezes o formato AN, onde N é o número de caracteres reservados para se escrever uma variável do tipo caracter. Por exemplo, na linha `12 FORMAT (2(A10), "= variáveis A e B no formato 2(A10)")` do programa usa-se duas vezes o formato A10 para se escrever duas variáveis do tipo caracter.
- 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 do DOS, mostrada na Figura 6.1, dentro da qual tem-se:
- Na primeira linha, o comentário Variáveis do tipo caracter, resultado do comando `WRITE(*,*) "Variáveis do tipo caracter"` do programa.
 - Na segunda linha, o comentário Entre com os conteúdos de A e B, resultado do comando `WRITE(*,*) "Entre com os conteúdos de A e B"` do programa.
 - Na terceira linha, o programa pára e fica aguardando que sejam fornecidos os conteúdos das variáveis A e B, resultado do comando `READ(*,*) A, B` do programa. Para que o programa

continue sua execução é necessário **digitar “Rio 45”, “Teste”**, por exemplo, e, em seguida, **clique na tecla enter**.

- d) Em seguida ocorrerá a criação do arquivo `saida6a.txt`, dentro do diretório do projeto, como resultado do comando `OPEN(9, file="saida6a.txt")` do programa, uma vez que este arquivo ainda não existe quando o programa `6a.f90` for executado pela primeira vez. Este comando também declara que será usado o número 9 no programa como referência ao arquivo `saida6a.txt`.
- e) Como resultado da linha `VER = SYSTEM("Notepad saida6a.txt")` do programa, o aplicativo Notepad abre automaticamente o arquivo `saida6a.txt` cujo conteúdo é mostrado na Figura 6.2. Na segunda e terceira linha do arquivo `saida6a.txt` o conteúdo da variável A (Rio 45) foi cortado parcialmente devido aos poucos caracteres usados para escrevê-la, no caso 2 e 5, definidos por A2 e A5. Para evitar este tipo de problema, sempre se deve superdimensionar o espaço para escrever o conteúdo de cada variável.

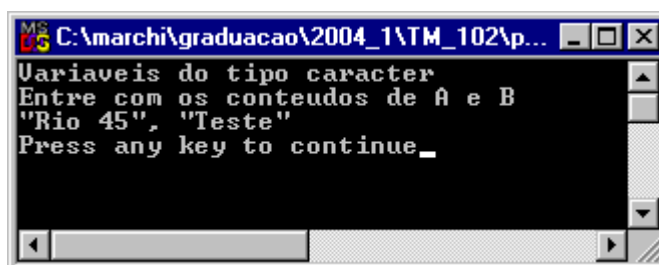


Figura 6.1 Janela DOS do programa `6a.f90`.

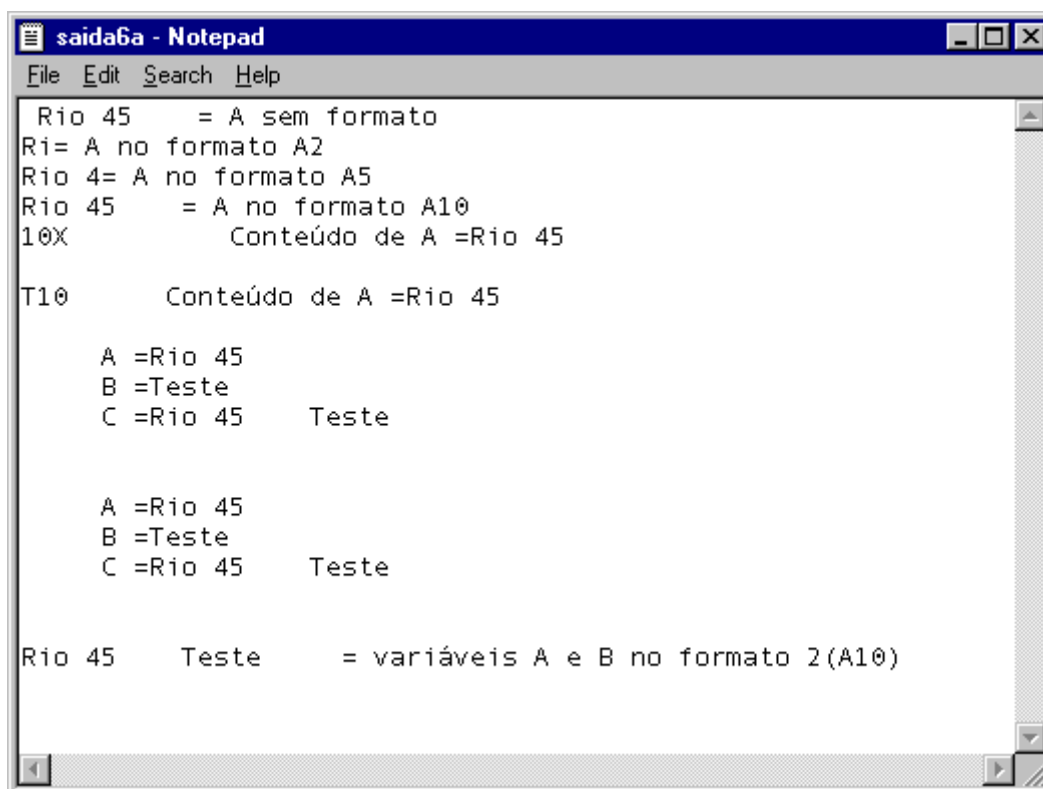


Figura 6.2 Arquivo `saida6a.txt` do programa `6a.f90`.

- 9) Até entender, **comparar** o resultado de cada comando WRITE, e seu respectivo comando FORMAT no programa6a.f90, mostrado em cada linha da Figura 6.2 ou no arquivo saida6a.txt.
- 10) **Executar** novamente o programa com outros conteúdos para as variáveis A e B. **Analisar** os novos resultados.

6.2 FORMATOS DE EDIÇÃO PARA VARIÁVEIS DO TIPO INTEIRO: programa6b.f90

- 1) Nesta seção será usado o mesmo 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 o programa-fonte do projeto.
- 2) No Fortran, **seguindo o procedimento-padrão**, **criar e inserir** no projeto o programa-fonte **programa6b.f90**
- 3) Dentro do espaço de edição do Fortran, na subjanela maior, **copiar** exatamente o texto em vermelho mostrado na **Tabela 6.2**.

Tabela 6.2 Programa6b.f90.

```
USE PORTLIB

INTEGER A, B, C
INTEGER VER

WRITE(*,*) "Variaveis do tipo inteiro"
WRITE(*,*) "Entre com os valores de A e B"
READ(*,*) A, B

C = A + B

OPEN(9, file="saida6b.txt")

WRITE(9,*) A, "= A sem formato"
WRITE(9,21) A
WRITE(9,22) A
WRITE(9,23) A
WRITE(9,24) A
WRITE(9,25) A
WRITE(9,10) A, B, C
WRITE(9,11) A, B, C
```

```

WRITE (9,9) A, B, C

CLOSE (9)

VER = SYSTEM("Notepad saida6b.txt")

21 FORMAT ( I2, "= A no formato I2" )
22 FORMAT ( I5, "= A no formato I5" )
23 FORMAT ( I10, "= A no formato I10" )
24 FORMAT("10X", 10X, "Valor de A =", I10, 1/)
25 FORMAT("T10", T10, "Valor de A =", I10, 1/)
10 FORMAT(5X, "A =", I10, 1/, 5X, "B =", I10, 1/, 5X, "C =", I10, 2/)
11 FORMAT( 5X, "A =", I10, 1/,      &
          5X, "B =", I10, 1/,      &
          5X, "C =", I10, 2/       )
 9 FORMAT ( 3(I10), "= variáveis A, B e C no formato 3(I10)" )

END

```

4) Comentários sobre o programa:

- a) O objetivo do programa6b.f90 é explicar alguns formatos de edição utilizados com variáveis do tipo inteiro.
- b) No caso do programa6b.f90, os comandos FORMAT foram todos aglomerados no final do programa-fonte, na mesma ordem de seus respectivos comandos WRITE.
- c) O identificador de variáveis do tipo inteiro é a letra I seguida de um número que especifica a quantidade máxima de algarismos ou colunas para escrever cada variável. Dentro deste espaço, o conteúdo de cada variável do tipo inteiro é escrito da direita para a esquerda. Uma coluna é usada para o sinal do número inteiro. Se o valor inteiro da variável ocupar mais espaço, colunas ou algarismos do que o reservado para escrevê-la, no lugar do número serão apresentados apenas asteriscos. Isso significa que o formato utilizado é incompatível com o valor da variável. Um exemplo de uso deste identificador é I5 na linha 22 `FORMAT (I5, "= A no formato I5")` do programa.
- d) Os identificadores X, para espaços em branco, T, para tabulação, e N/, para o cursor descer N linhas, explicados na seção 6.1 para variáveis do tipo caracter, também se aplicam com variáveis do tipo inteiro, conforme pode-se ver no programa6b.f90.
- e) O formato M(IN) é equivalente a M vezes o formato IN, onde N é o número de colunas ou algarismos reservados para se escrever uma variável do tipo inteiro. Por exemplo, na linha 9

`FORMAT (3(I10), "= variáveis A, B e C no formato 3(I10)")` do programa usa-se três vezes o formato I10 para se escrever três variáveis do tipo inteiro.

- 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 do DOS, mostrada na Figura 6.3, dentro da qual tem-se:
 - a) Na primeira linha, o comentário Variáveis do tipo inteiro, resultado do comando `WRITE(*,*) "Variaveis do tipo inteiro"` do programa.
 - b) Na segunda linha, o comentário Entre com os valores de A e B, resultado do comando `WRITE(*,*) "Entre com os valores de A e B"` do programa.
 - c) Na terceira linha, o programa pára e fica aguardando que sejam fornecidos os valores das variáveis A e B, resultado do comando `READ(*,*) A, B` do programa. Para que o programa continue sua execução é necessário **digitar 111 222**, por exemplo, e, em seguida, **clique na tecla enter**.
 - d) Em seguida ocorrerá a criação do arquivo saida6b.txt, dentro do diretório do projeto, como resultado do comando `OPEN(9, file="saida6b.txt")` do programa, uma vez que este arquivo ainda não existe quando o programa6b.f90 for executado pela primeira vez. Este comando também declara que será usado o número 9 no programa como referência ao arquivo saida6b.txt.
 - e) Como resultado da linha `VER = SYSTEM("Notepad saida6b.txt")` do programa, o aplicativo Notepad abre automaticamente o arquivo saida6b.txt cujo conteúdo é mostrado na Figura 6.4. Na segunda linha do arquivo saida6b.txt o valor da variável A (111) foi trocado por dois asteriscos. Isso ocorre porque o formato usado (I2) é incompatível para se escrever o valor da variável A, que neste caso precisa de pelo menos três colunas, ou seja, o formato mínimo seria I3. Para evitar este tipo de problema, sempre se deve superdimensionar o espaço para escrever o valor de cada variável.

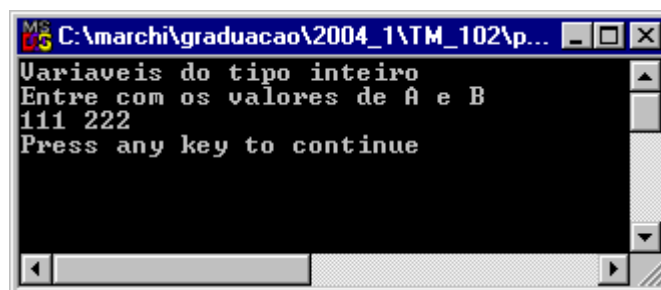
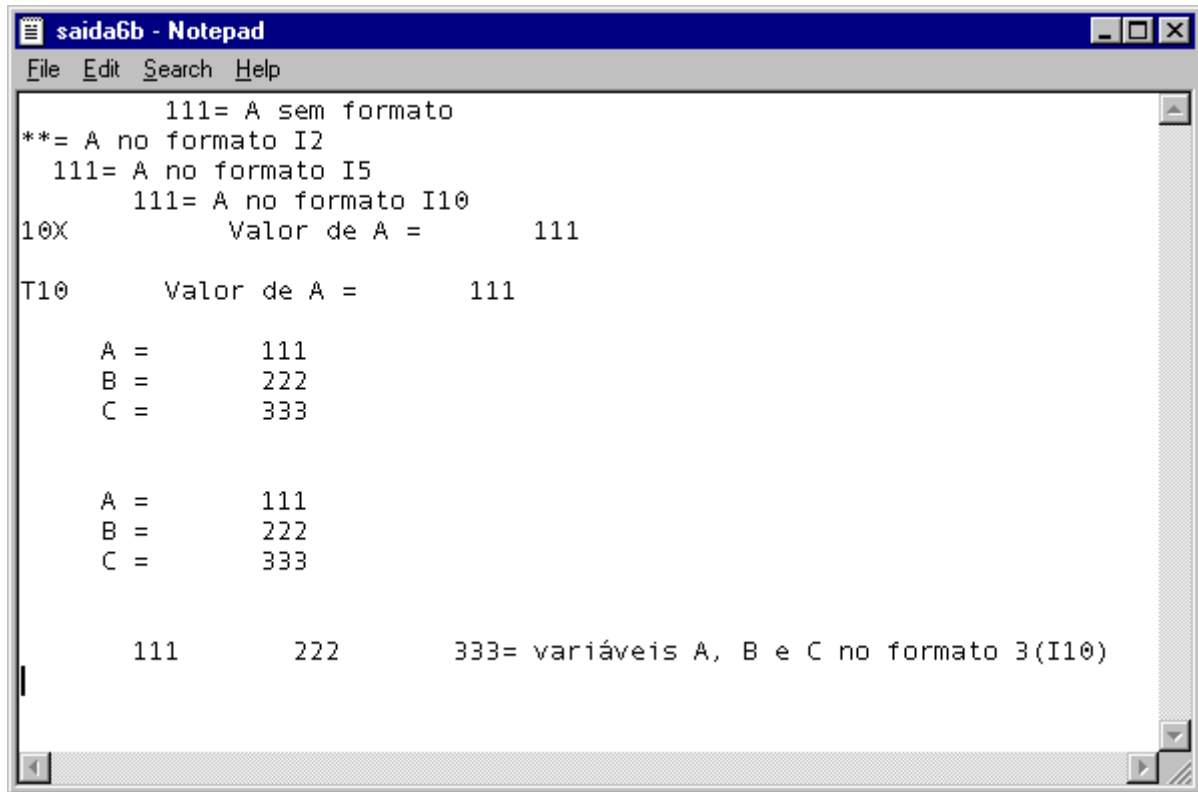


Figura 6.3 Janela DOS do programa6b.f90.

- 8) Até entender, **comparar** o resultado de cada comando WRITE, e seu respectivo comando FORMAT no programa6b.f90, mostrado em cada linha da Figura 6.4 ou no arquivo saida6b.txt.

- 9) **Executar** novamente o programa com outros valores para as variáveis A e B. **Analisar** os novos resultados.



```
saida6b - Notepad
File Edit Search Help
      111= A sem formato
**= A no formato I2
      111= A no formato I5
      111= A no formato I10
10X      Valor de A =      111

T10      Valor de A =      111

      A =      111
      B =      222
      C =      333

      A =      111
      B =      222
      C =      333

      111      222      333= variáveis A, B e C no formato 3(I10)
```

Figura 6.4 Arquivo saida6b.txt do programa6b.f90.

6.3 FORMATOS DE EDIÇÃO PARA VARIÁVEIS DO TIPO REAL: programa6c.f90

- 1) Nesta seção será usado o mesmo 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 o programa-fonte do projeto.
- 2) No Fortran, **segundo o procedimento-padrão**, **criar e inserir** no projeto o programa-fonte **programa6c.f90**
- 3) Dentro do espaço de edição do Fortran, na subjanela maior, **copiar** exatamente o texto em vermelho mostrado na **Tabela 6.3**.
- 4) Comentários sobre o programa:
 - a) O objetivo do programa6c.f90 é explicar alguns formatos de edição utilizados com variáveis do tipo real.
 - b) No caso do programa6c.f90, os comandos FORMAT foram todos aglomerados no início do programa-fonte, antes mesmo de seus respectivos comandos WRITE.
 - c) Os dois principais identificadores de variáveis do tipo real são representados pelas letras F e E.

d) O formato de ponto flutuante é usado para números reais. Ele é denotado por FX.Y, onde Y é um número que equivale à quantidade de algarismos usados para representar a parte decimal do valor da variável. X é um número que equivale à quantidade total de colunas usadas para representar o valor da variável, incluindo seu sinal, o ponto decimal e a parte decimal. O número X sempre tem que ser maior do que Y. No programa, por exemplo, são usados os formatos F5.2 e F5.0.

Tabela 6.3 Programa6c.f90.

```
USE PORTLIB

REAL A, B, C
INTEGER VER

11 FORMAT( 5X, "A =", 1PE15.3, 1/,      &
          5X, "B =", 1PE15.3, 1/,      &
          5X, "C =", 1PE15.3, 1/      )
12 FORMAT ( 3(1PE15.3), "= variáveis A, B e C no formato 3(1PE15.3)" )
21 FORMAT ("A no formato F5.2   =", F5.2,   1/, &
          "A no formato F5.0   =", F5.0,   1/, &
          "A no formato E15.3  =", E15.3,  1/, &
          "A no formato 1PE15.3 =", 1PE15.3, 1/ )

WRITE(*,*) "Variaveis do tipo real"
WRITE(*,*) "Entre com os valores de A e B"
READ(*,*) A, B

C = A + B

OPEN(9, file="saida6c.txt")

WRITE(9,*) A, "= A sem formato"
WRITE(9,21) A, A, A, A
WRITE(9,11) A, B, C
WRITE(9,12) A, B, C

CLOSE(9)

VER = SYSTEM("Notepad saida6c.txt")

END
```

- e) O formato exponencial também é usado para números reais. Ele é denotado por EX.Y, onde Y é um número que equivale à quantidade de algarismos usados para representar a parte decimal do valor da variável. X é um número que equivale à quantidade total de colunas usadas para representar o valor da variável, incluindo seu sinal, o ponto decimal, a parte decimal, a letra E, o sinal do expoente e dois algarismos para o valor do expoente. O número X sempre tem que ser maior do que Y. No programa, por exemplo, é usado o formato E15.3.
- f) No formato EX.Y, nenhum algarismo é usado na parte inteira. Para que seja usado um algarismo na parte inteira do número real, pode-se usar o formato 1PEX.Y. No programa, por exemplo, é usado o formato 1PE15.3.
- g) Dentro das X colunas dos formatos FX.Y, EX.Y e 1PEX.Y, o conteúdo de cada variável do tipo real é escrito da direita para a esquerda. Os valores de X e Y são definidos pelo programador.
- h) Os identificadores X, para espaços em branco, T, para tabulação, e N/, para o cursor descer N linhas, explicados na seção 6.1 para variáveis do tipo caracter, também se aplicam com variáveis do tipo real, conforme pode-se ver no programa6c.f90.
- i) O formato M(1PEX.Y) é equivalente a M vezes o formato 1PEX.Y. Por exemplo, na linha 12
`FORMAT (3(1PE15.3), "= variáveis A, B e C no formato 3(1PE15.3)")` do programa usa-se três vezes o formato 1PE15.3 para se escrever três variáveis do tipo real.
- 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 do DOS, mostrada na Figura 6.5, dentro da qual tem-se:
- a) Na primeira linha, o comentário Variáveis do tipo real, resultado do comando `WRITE(*,*)`
`"Variáveis do tipo real"` do programa.
- b) Na segunda linha, o comentário Entre com os valores de A e B, resultado do comando `WRITE(*,*)` `"Entre com os valores de A e B"` do programa.
- c) Na terceira linha, o programa pára e fica aguardando que sejam fornecidos os valores das variáveis A e B, resultado do comando `READ(*,*)` `A, B` do programa. Para que o programa continue sua execução é necessário **digitar -1.234 2.22**, por exemplo, e, em seguida, **clique na tecla enter**.
- d) Em seguida ocorrerá a criação do arquivo saída6c.txt, dentro do diretório do projeto, como resultado do comando `OPEN(9, file="saida6c.txt")` do programa, uma vez que este arquivo ainda não existe quando o programa6c.f90 for executado pela primeira vez. Este comando também declara que será usado o número 9 no programa como referência ao arquivo saída6c.txt.
- e) Como resultado da linha `VER = SYSTEM("Notepad saída6c.txt")` do programa, o aplicativo Notepad abre automaticamente o arquivo saída6c.txt cujo conteúdo é mostrado na Figura 6.6.

Dependendo do formato empregado, alguns algarismos do número real são truncados, como se pode ver da segunda à quarta linha do arquivo saida6c.txt.

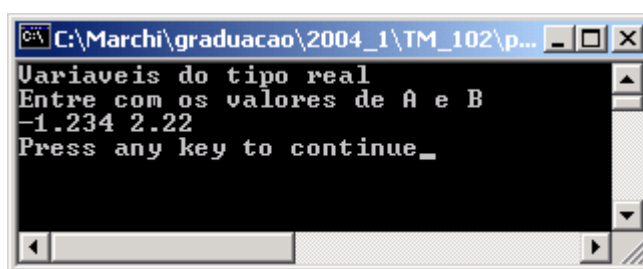


Figura 6.5 Janela DOS do programa6c.f90.

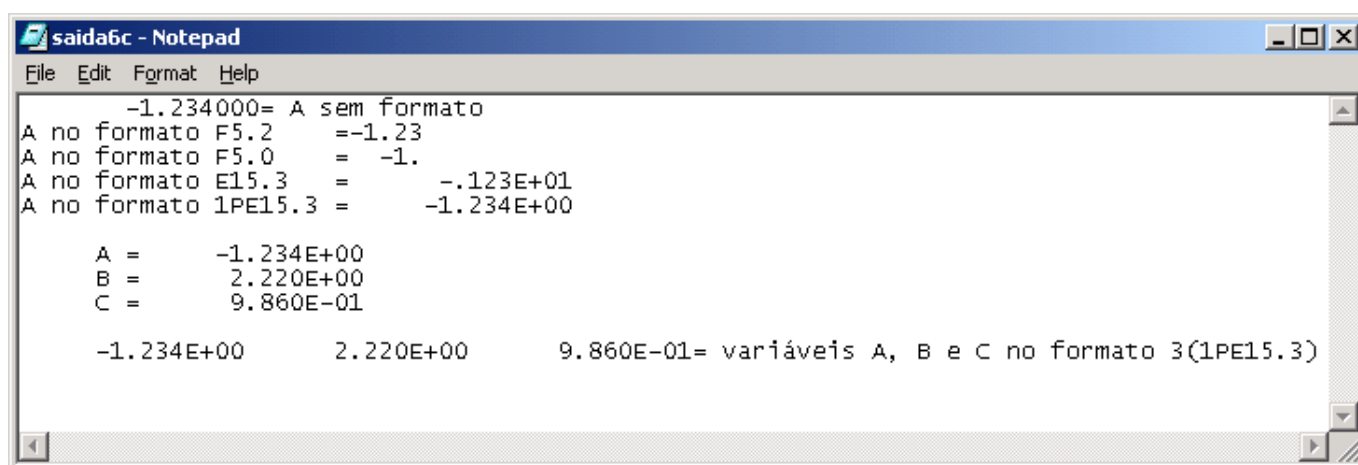


Figura 6.6 Arquivo saida6c.txt do programa6c.f90.

- 8) Até entender, **comparar** o resultado de cada comando WRITE, e seu respectivo comando FORMAT no programa6c.f90, mostrado em cada linha da Figura 6.6 ou no arquivo saida6c.txt.
- 9) **Executar** novamente o programa com outros valores para as variáveis A e B. **Analisar** os novos resultados.
- 10) Num mesmo comando FORMAT, podem ser usados identificados de variáveis do tipo inteiro, caracter e real, além de comentários, e dos formatos NX, N/ e TN.

6.4 RESUMO DE REGRAS GERAIS DE EDIÇÃO DE PROGRAMAS EM FORTRAN

- 1) Em cada linha do programa, tudo que estiver após o símbolo de exclamação (!) não é executado pelo programa. Serve apenas como comentário usado para esclarecer o que faz cada parte do programa. Dentro do Fortran PowerStation 4.0, todos os comentários ficam na cor verde. Linhas em branco são equivalentes a ter o símbolo (!) de comentário na primeira coluna de cada linha em branco.
- 2) Tudo que é criado ou definido pelo programador fica na cor preta dentro do Fortran PowerStation 4.0.

- 3) Todos os comandos intrínsecos da linguagem FORTRAN ficam na cor azul dentro do Fortran PowerStation 4.0.
- 4) O símbolo & é usado ao final de uma linha de programa para indicar que o comando iniciado numa linha contínua na linha seguinte.
- 5) Cada linha de programa pode usar até 132 colunas, incluindo os espaços em branco.
- 6) Em geral, espaços em branco têm apenas função estética no programa, não afetando os comandos ou as instruções do programador. Os espaços em branco são considerados quando usados dentro de comentários nos comandos WRITE e FORMAT.
- 7) Nos comandos da linguagem FORTRAN e em nomes de variáveis, não existe distinção entre letras maiúsculas ou minúsculas. Por exemplo, WRITE, write, Write e wRiTe são equivalentes. A distinção entre letras minúsculas e maiúsculas existe dentro de comentários nos comandos WRITE e FORMAT.
- 8) Na linguagem FORTRAN, os nomes de variáveis podem ter até 31 caracteres que incluem combinações das 26 letras do alfabeto, dos algarismos 0, 1, 2, 3 a 9, e o traço de sublinhado. Não é permitido acentuação e todas as variáveis devem começar por uma letra. Exemplos de nomes de variáveis (separadas por vírgula): v, va, variavel, variavel_2, variavel_31, nome03, nome_03, nome_da_rua.

6.5 SEQUÊNCIA OBRIGATÓRIA DE COMANDOS EM FORTRAN

Até este capítulo foram vistos comandos da linguagem FORTRAN que podem ser divididos em cinco tipos:

- 1) Comando para inclusão de biblioteca no programa-fonte: USE
- 2) Comandos para declarar tipos de variáveis: INTEGER, REAL e CHARACTER
- 3) Outros comandos intrínsecos (WRITE, READ, OPEN, CLOSE) ou não do FORTRAN (SYSTEM), funções intrínsecas (LOG, TRIM etc) e qualquer outro comando criado pelo programador, por exemplo, cálculos e operações com strings
- 4) Comando de encerramento do programa-fonte: END
- 5) Comando de edição: FORMAT

Se todos os cinco tipos de comandos forem usados simultaneamente num programa-fonte, a seqüência obrigatória entre eles é exatamente a apresentada acima, exceto para o quinto tipo. O comando FORMAT pode ficar em qualquer lugar do programa-fonte entre os comandos USE e END.

Para os comandos do tipo 2 (INTEGER, REAL e CHARACTER), a ordem entre eles pode ser qualquer. O mesmo se aplica para os comandos do tipo 3, exceto com arquivos nos quais para um arquivo específico a seqüência obrigatória é: OPEN, WRITE e outros comandos, CLOSE.

Encerrar a sessão seguindo o [procedimento-padrão](#).

6.6 EXERCÍCIOS

Exercício 6.1

Editar um programa-fonte em FORTRAN para realizar um cálculo de interpolação linear executando o seguinte algoritmo (passos):

- 1) Ler os valores de duas variáveis do tipo real, chamadas X1 e X2, que correspondem a dois dados tabelados de uma variável independente
- 2) Ler os valores de duas variáveis do tipo real, chamadas Y1 e Y2, que correspondem a dois dados tabelados de uma variável dependente de X
- 3) Ler o valor de uma variável do tipo real, chamada X, para o qual se deseja realizar a interpolação
- 4) Calcular o valor da variável Y, em função do valor de X, usando interpolação linear
- 5) Utilizando comandos FORMAT, escrever no arquivo chamado saida_6p1.txt os cinco valores lidos e o resultado de Y, juntamente com comentários para identificá-los
- 6) Executar o programa para $X1 = 10$, $X2 = 20$, $Y1 = 100$, $Y2 = -20$ e $X = 12$. O resultado deve ser $Y = 76$.

Exercício 6.2

Editar um programa-fonte em FORTRAN para realizar cálculos com uma progressão geométrica (P.G.) executando o seguinte algoritmo (passos):

- 1) Ler o primeiro valor (real) da P.G., denotado por a_1
- 2) Ler a razão (número real) entre dois termos subseqüentes da P.G., denotada por q
- 3) Ler o número (inteiro) de termos da P.G., denotado por N
- 4) Calcular o valor (real) do último termo da P.G., denotado por a_n
- 5) Calcular a soma de todos os termos da P.G., denotado por S_n
- 6) Utilizando comandos FORMAT, escrever no arquivo chamado saida_6p2.txt os três valores lidos e os dois calculados, juntamente com comentários para identificá-los
- 7) Executar o programa para $a_1 = 1.5$, $q = 1.1$ e $N = 20$. Os resultados devem ser aproximadamente $a_n \approx 9.174$ e $S_n \approx 85.91$.