

OBJETIVOS DO CAPÍTULO

- Conceitos de: arquivo de saída, biblioteca, funções matemáticas intrínsecas
- Criar e usar arquivos para mostrar resultados da execução de um programa em FORTRAN
- Usar o aplicativo Notepad do Windows para mostrar o conteúdo de um arquivo durante a execução de um programa em FORTRAN
- Funções matemáticas intrínsecas do FORTRAN
- Comandos do FORTRAN: WRITE(número,*), OPEN, CLOSE, USE, SYSTEM

5.1 programa5a.f90

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

- 1) No Windows: **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 **programa05** 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 **programa5a.f90**
- 4) Conforme é mostrado na Figura 5.1, dentro do espaço de edição do Fortran, na subjanela maior, **copiar** exatamente o texto em vermelho mostrado na **Tabela 5.1**.
- 5) Comentários sobre o programa:
 - a) Até aqui o comando WRITE foi utilizado com dois asteriscos dentro do parênteses, isto é, foi usado na forma WRITE(*,*). O primeiro asterisco do comando WRITE representa o “local” ou dispositivo que é usado para “escrever” algum comentário ou resultado do programa. Este asterisco pode ser substituído por um número. O número 6 representa uma janela do DOS, como aquela que vem sendo usada até aqui para apresentar os resultados dos programas; por exemplo, a janela mostrada na Figura 5.2. Portanto, o uso de asterisco no comando WRITE equivale a empregar o número 6.
 - b) A única diferença entre o programa5a.f90 e o programa3c.f90, da seção 3.1 do capítulo 3, é o uso do número 6 no lugar do primeiro asterisco dos comandos WRITE.
- 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 5.2, dentro da qual tem-se:
 - a) Na primeira linha, o comentário Entre com o valor de A =, resultado do comando **WRITE(6,*)** "Entre com o valor de A =" do programa.

Tabela 5.1 Programa5a.f90.

```

REAL A
WRITE(6,*) "Entre com o valor de A ="
READ(*,*) A
WRITE(6,*) "Valor de A = ", A
END

```

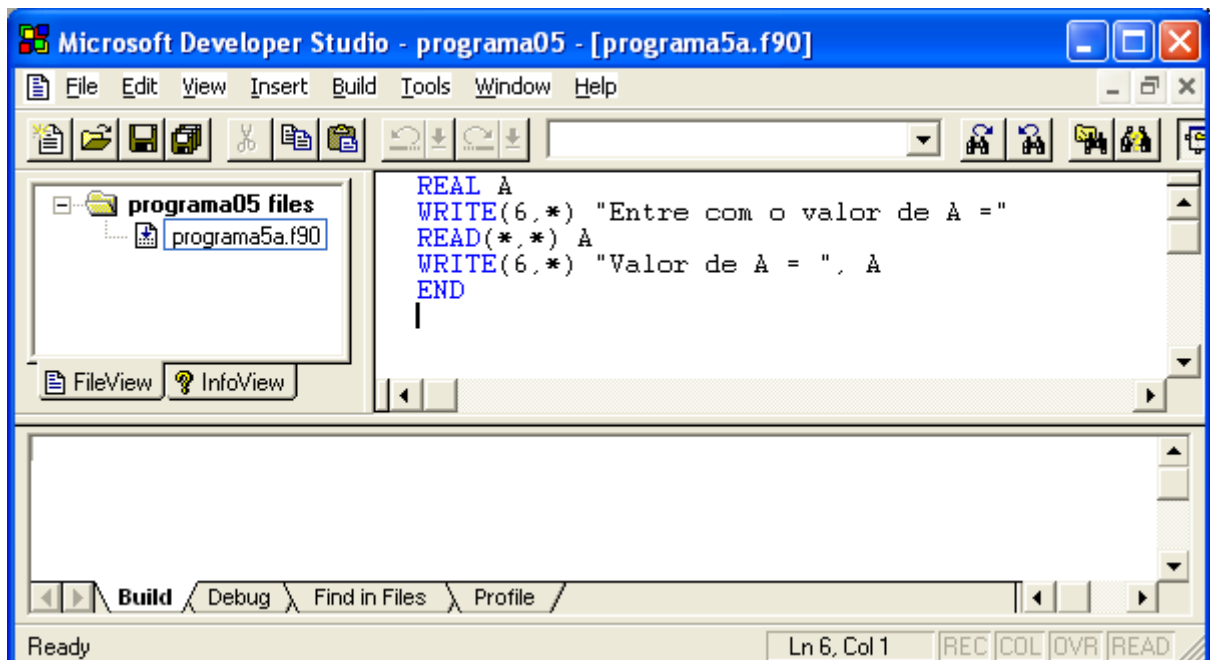


Figura 5.1 Programa5a.f90.

- 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, **clique na tecla enter**.
- c) Na terceira linha, o comentário Valor de A = e o valor da variável A, resultado do comando **WRITE(6,*) "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.
- 9) O resultado da execução do programa5a.f90 é exatamente igual a do programa3c.f90, da seção 3.1 do capítulo 3. Conforme explicado no item 5a, acima, isso ocorre devido à equivalência completa entre os comandos WRITE(*,*) e WRITE(6,*).

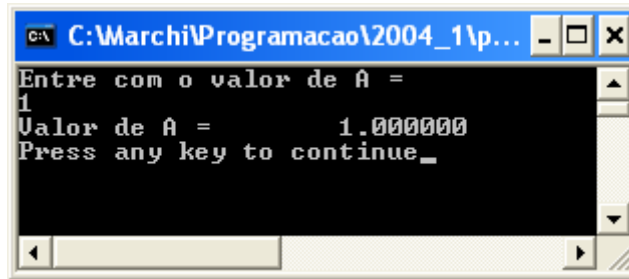


Figura 5.2 Resultado da execução do programa5a.f90.

5.2 programa5b.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 apresentado na seção 1.7 do capítulo 1, **criar e inserir** no projeto chamado programa05 o programa-fonte **programa5b.f90**
- 3) Dentro do espaço de edição do Fortran, na subjanela maior, **copiar** exatamente o texto em vermelho mostrado na **Tabela 5.2**.

Tabela 5.2 Programa5b.f90.

```
REAL A
WRITE(6,*) "Entre com o valor de A ="
READ(*,*) A

OPEN(7, file="saida5b.txt")
WRITE(7,*) "Valor de A = ", A
CLOSE(7)

END
```

- 4) Comentários sobre o programa:

- a) No programa5b.f90 são usados dois novos comandos do FORTRAN: OPEN e CLOSE. Ambos devem ser usados em conjunto.
 - b) A forma do comando OPEN que será usada neste texto é OPEN(número, file="nome"). O comando OPEN é usado para: definir um número que será usado como referência a um arquivo; e para abrir o arquivo chamado "nome". O número pode ser de 1 a 99 e 105 a 299. A utilização deste número num comando WRITE(número,*) resulta na escrita de comentários ou variáveis no arquivo que corresponde ao número definido no comando OPEN. O nome do arquivo pode ser qualquer um mas deve ficar entre aspas. Quando se declara um nome de arquivo num comando OPEN, se o arquivo não existir, ele é criado dentro do diretório do projeto.
 - c) A forma do comando CLOSE que será usada neste texto é CLOSE(número). O comando CLOSE é usado para fechar ou encerrar o uso de um arquivo aberto anteriormente com o comando OPEN.
 - d) Todos arquivos abertos com o comando OPEN são fechados ao final da execução do programa, existindo ou não o comando CLOSE.
 - e) Dentro de um programa, para um mesmo número ou arquivo, a seqüência obrigatória dos comandos é a seguinte: OPEN, WRITE, CLOSE. Para um mesmo arquivo, deve existir apenas um comando OPEN e um CLOSE dentro do programa, e podem existir diversos comandos WRITE.
 - f) Podem ser usados vários arquivos simultaneamente num programa.
 - g) A principal vantagem de se usar um arquivo é ter um registro permanente dos resultados do programa para posterior análise. Isso fica mais evidente quanto maior for a quantidade de resultados de um programa.
- 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 5.3, dentro da qual tem-se:

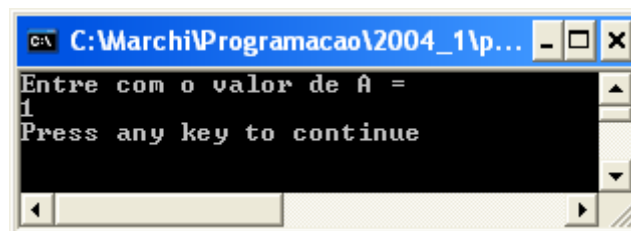


Figura 5.3 Resultado da execução do programa5b.f90 na janela DOS com A = 1.

- 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 conteúdo 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, **clique na tecla enter**.

- c) Em seguida ocorrerá a criação do arquivo `saida5b.txt`, dentro do diretório do projeto, como resultado do comando `OPEN(7, file="saida5b.txt")` do programa, uma vez que este arquivo ainda não existe quando o programa `programa5b.f90` for executado pela primeira vez. Este comando também declara que será usado o número 7 no programa como referência ao arquivo `saida5b.txt`.
- d) Conforme é mostrado na Figura 5.4, na primeira linha do arquivo `saida5b.txt` está escrito o comentário `Valor de A =` e o valor da variável real A como resultado do comando `WRITE(7,*) "Valor de A = ", A` do programa.
- e) O arquivo `saida5b.txt` é fechado, isto é, nada mais pode ser escrito nele após o comando `CLOSE(7)` do programa.

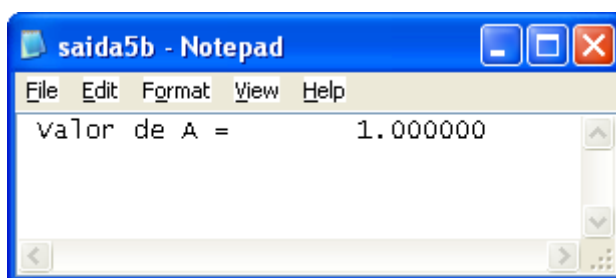


Figura 5.4 Resultado da execução do programa `programa5b.f90` no arquivo `saida5b.txt` com $A = 1$.

- 8) A existência do arquivo `saida5b.txt` pode ser comprovada ao se executar **Start, Programs, Windows NT Explorer**. Em seguida, **indicar o diretório do projeto**, no caso, `programa05`. Finalmente, ao se **clique duas vezes sobre o nome do arquivo**, ele será aberto, podendo-se ver o seu conteúdo.
- 9) Executar novamente o programa, através de **Build, Execute**, com $A = -123$. Neste caso, como já existe o arquivo `saida5b.txt` ele será usado para escrever o novo resultado do programa sobre o resultado anterior, conforme é mostrado na Figura 5.5. Ou seja, o resultado anterior, mostrado na Figura 5.4 é eliminado ou apagado. Para evitar isso, é necessário gravar os resultados em arquivos diferentes a cada execução do programa. Outra possibilidade é alterar o nome do arquivo de resultados antes do programa ser executado novamente. **Executar** o indicado no item anterior (8) para ver o novo conteúdo do arquivo `saida5b.txt`.
- 10) No comando OPEN também é possível especificar todo o caminho ou o diretório aonde o arquivo deve ser criado ou aberto. Quando o diretório não é especificado, por default, usa-se o diretório do projeto ou o diretório no qual o programa é executado. Por exemplo, **trocar** a linha `OPEN(7, file="saida5b.txt")`, pelo seguinte: `OPEN(7, file="C:\Temp\saida5b.txt")`. **Compilar** novamente o programa `programa5b.f90`, **gerar** o seu executável, **executar** o programa. Usar o aplicativo

Windows NT Explorer para **encontrar** e **abrir** o arquivo saida5b.txt, que agora deverá estar no diretório C:\Temp.

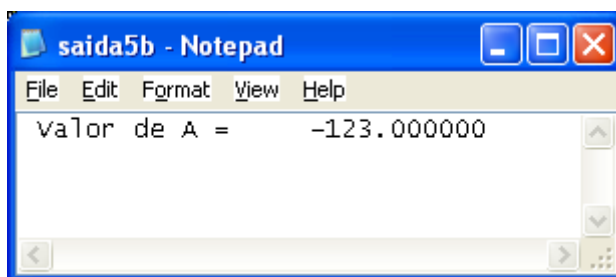


Figura 5.5 Resultado da execução do programa5b.f90 no arquivo saida5b.txt com $A = -123$.

11) Na próxima seção, é apresentada uma forma de se automatizar a abertura do arquivo de saída. Isso permite maior rapidez na visualização dos resultados de um programa.

5.3 programa5c.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 apresentado na seção 1.7 do capítulo 1, **criar e inserir** no projeto chamado programa05 o programa-fonte **programa5c.f90**
- 3) Dentro do espaço de edição do Fortran, na subjanela maior, **copiar** exatamente o texto em vermelho mostrado na **Tabela 5.3**.
- 4) Comentários sobre o programa:
 - a) No programa5c.f90 são usados dois novos comandos do FORTRAN: USE e SYSTEM.
 - b) O comando USE biblio é usado para incluir dentro do programa uma “biblioteca” chamada biblio, onde biblio deve ser substituído pelo nome da biblioteca desejada. Uma “biblioteca”, no presente contexto, é uma coleção de comandos que não existem automaticamente dentro da linguagem FORTRAN. Todos os comandos usados até aqui, por exemplo WRITE, já existem automaticamente dentro da linguagem FORTRAN, mas o comando SYSTEM, não. Para ele ser usado é necessário incluir no programa a biblioteca aonde este comando está definido, no caso é a biblioteca chamada PORTLIB.
 - c) A linguagem FORTRAN permite a qualquer programador criar suas próprias bibliotecas, como será visto nos capítulos avançados deste texto.

Tabela 5.3 Programa5c.f90.

```

USE PORTLIB

REAL A
INTEGER VER

WRITE(6,*) "Entre com o valor de A ="
READ(*,*) A
OPEN(7, file="saida5c.txt")
WRITE(7,*) "Valor de A = ", A
CLOSE(7)
VER = SYSTEM ("Notepad saida5c.txt")

END

```

- d) No editor do aplicativo Fortran, o comando SYSTEM não fica na cor azul por não ser um dos comandos básicos da linguagem FORTRAN. O comando SYSTEM é utilizado para executar comandos que interagem com o sistema operacional Windows. Isso permite que, durante a execução do programa escrito em FORTRAN, seja realizado o seguinte: executar outros programas ou aplicativos; criar ou deletar diretórios ou arquivos; e qualquer outro comando possível de se executar numa janela do tipo DOS. Para utilizar o comando SYSTEM deve-se definir uma variável do tipo inteiro. Depois, usar esta variável na linha do programa aonde se deseja executar o comando SYSTEM. Dentro de parênteses e aspas, deve-se incluir a seqüência de comandos que se deseja executar no Windows.
- e) A linha **USE PORTLIB** declara que a biblioteca chamada PORTLIB será acrescentada ao programa.
- f) A linha **VER = SYSTEM ("Notepad saida5c.txt")** executa o comando SYSTEM para que o aplicativo NOTEPAD do Windows abra o arquivo chamado saida5c.txt. A variável VER, usada nesta linha, foi definida como uma variável do tipo inteiro na linha **INTEGER VER** do programa.
- 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**, ocorrerá exatamente o mesmo já explicado na seção anterior para o programa5b.f90. A única diferença é que quase no final da execução do programa, o arquivo de resultados, que foi denominado de saida5c.txt, será aberto automaticamente pelo aplicativo Notepad para mostrar o seu conteúdo. Ao se fechar o Notepad, a execução do programa5c.f90 será encerrada.
- 8) No caso do comando OPEN ser usado para especificar todo o caminho ou o diretório aonde o arquivo deve ser criado ou aberto, o comando SYSTEM também deve ser adaptado. Por exemplo, se o arquivo

saida5c.txt for criado no diretório C:\Temp com o comando `OPEN(7, file="C:\Temp\saida5c.txt")`, o comando correspondente para abrir automaticamente o arquivo `saida5c.txt` é `VER = SYSTEM ("Notepad C:\Temp\saida5c.txt")`.

5.4 programa5d.f90

- 1) Nesta seção será usado o mesmo projeto da seção anterior deste capítulo. 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 apresentado na seção 1.7 do capítulo 1, **criar e inserir** no projeto chamado programa05 o programa-fonte **programa5d.f90**
- 3) Dentro do espaço de edição do Fortran, na subjanela maior, **copiar** exatamente o texto em vermelho mostrado na **Tabela 5.4**.
- 4) Comentários sobre o programa:
 - a) Conforme pode-se ver na Figura 5.6, ABS, LOG10, ACOS, COS e COSH estão escritos em azul dentro do Fortran. Elas são chamadas de funções intrínsecas do FORTRAN, ou seja, são funções ou comandos que existem dentro da linguagem FORTRAN. Elas são funções matemáticas usadas em cálculos que envolvem variáveis do tipo real. Outras funções são apresentadas nas Tabelas 5.5 e 5.6.
 - b) 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.
- 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 5.7, 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 -10 para a variável A**, por exemplo, e, em seguida, **clicar na tecla enter**.

- c) Em seguida ocorrerá a criação do arquivo `saida5d.txt`, dentro do diretório do projeto, como resultado do comando `OPEN(8, file="saida5d.txt")` do programa, uma vez que este arquivo ainda não existe quando o programa `programa5d.f90` for executado pela primeira vez. Este comando também declara que será usado o número 8 no programa como referência ao arquivo `saida5d.txt`.
- d) Os resultados das funções aplicadas no programa `programa5d.f90` são mostrados na Figura 5.8.

Tabela 5.4 Programa5d.f90.

```
USE PORTLIB

REAL A, B, C, Pi
INTEGER VER

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

OPEN(8, file="saida5d.txt")

WRITE(8,*) "Valor de A = ", A

B = ABS(A)
WRITE(8,*) "B = Módulo de A =", B

C = LOG10(B)
WRITE(8,*) "C: Logaritmo decimal de B =", C

Pi = ACOS(-1.0)
WRITE(8,*) "Pi =", Pi

D = COS(Pi)
WRITE(8,*) "Cosseno de Pi =", D

D = COSH(Pi)
WRITE(8,*) "Cosseno hiperbólico de Pi =", D

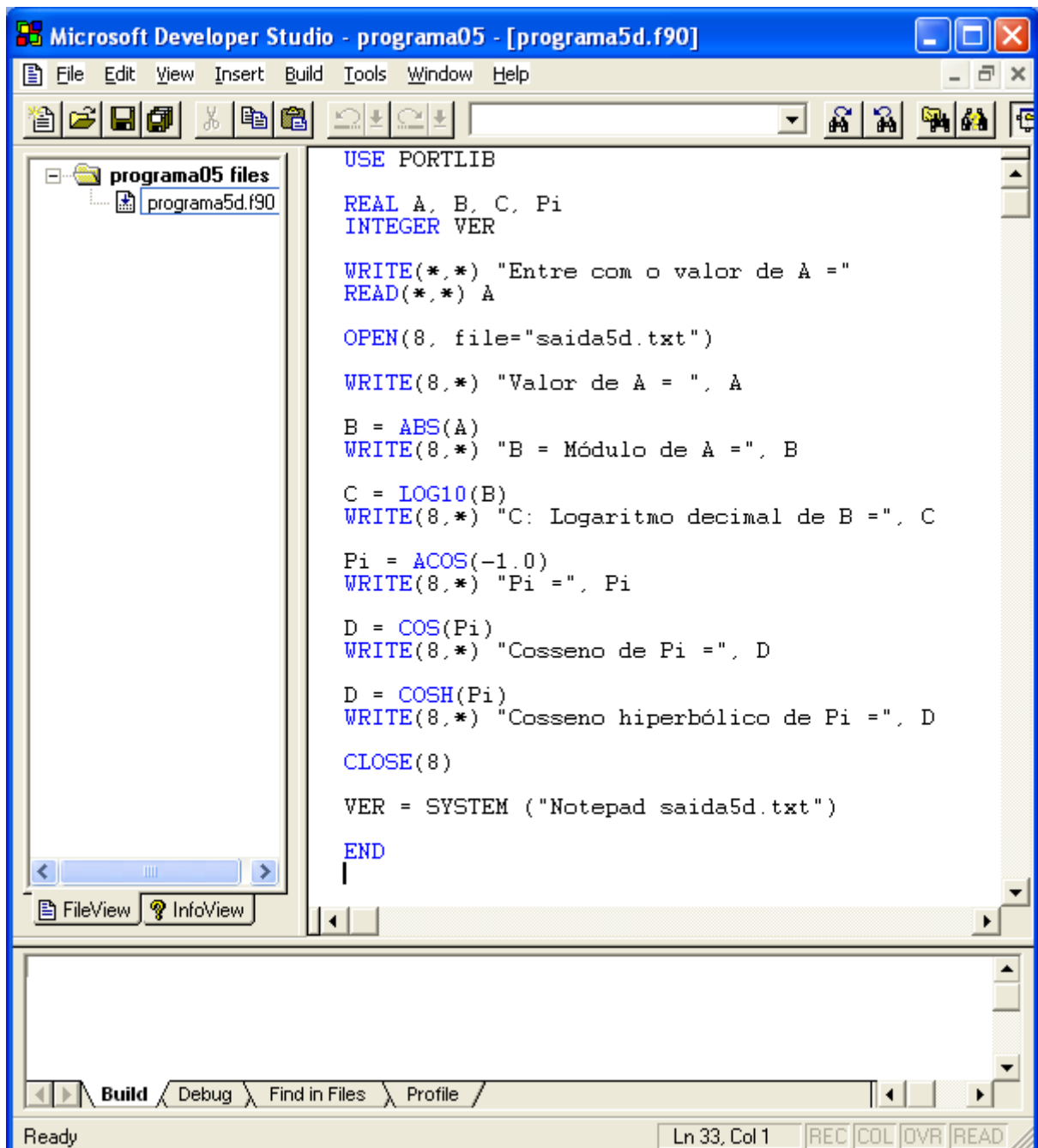
CLOSE(8)

VER = SYSTEM ("Notepad saida5d.txt")

END
```

- 8) **Comparar** cada resultado mostrado na Figura 5.8 com o resultado obtido de uma calculadora.

- 9) **Executar** novamente o programa com outro valor para a variável A. **Comparar** cada novo resultado com o obtido de uma calculadora.
- 10) Os programas-executáveis também podem ser executados, rodados ou postos para funcionar fora do aplicativo Fortran. Para testar isso, primeiro, **fechar o aplicativo Fortran.** Em seguida, utilizando o aplicativo Windows NT Explorer, **copiar** o programa-executável chamado **programa05.exe** ou **programa05.application** para o diretório C:\Temp. Para executar o programa, basta **clique duas vezes sobre o arquivo copiado.** Como resultado, surgirá o arquivo saída5d.txt. O diretório C:\Temp é apenas um exemplo. O programa-executável pode ser copiado para qualquer outro diretório e ser executado lá.



```
USE PORTLIB
REAL A, B, C, Pi
INTEGER VER

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

OPEN(8, file="saida5d.txt")

WRITE(8,*) "Valor de A = ", A

B = ABS(A)
WRITE(8,*) "B = Módulo de A =", B

C = LOG10(B)
WRITE(8,*) "C: Logaritmo decimal de B =", C

Pi = ACOS(-1.0)
WRITE(8,*) "Pi =", Pi

D = COS(Pi)
WRITE(8,*) "Cosseno de Pi =", D

D = COSH(Pi)
WRITE(8,*) "Cosseno hiperbólico de Pi =", D

CLOSE(8)

VER = SYSTEM ("Notepad saida5d.txt")

END
```

Figura 5.6 Programa5d.f90.

Tabela 5.5 Algumas funções matemáticas do FORTRAN.

Função	Comando	Observação
Módulo ou valor absoluto	$Y = \text{ABS}(X)$	
Raiz quadrada	$Y = \text{SQRT}(X)$	
Exponencial: $Y = e^X$	$Y = \text{EXP}(X)$	
Logaritmo natural	$Y = \text{LOG}(X)$	
Logaritmo decimal	$Y = \text{LOG10}(X)$	
Sinal	$Y = \text{SIGN}(X,Z)$	Y = valor absoluto de X vezes o sinal de Z
Mínimo	$Y = \text{MIN}(X,Z,W)$	Y = valor mínimo entre X, Z e W; o argumento pode conter 2 ou mais variáveis
Máximo	$Y = \text{MAX}(X,Z,W)$	Y = valor máximo entre X, Z e W; o argumento pode conter 2 ou mais variáveis

Tabela 5.6 Algumas funções trigonométricas do FORTRAN.

Função	Comando	Observação
Seno	$Y = \text{SIN}(X)$	X em radianos
Cosseno	$Y = \text{COS}(X)$	X em radianos
Tangente	$Y = \text{TAN}(X)$	X em radianos
Arco-seno	$Y = \text{ASIN}(X)$	Y em radianos
Arco-cosseno	$Y = \text{ACOS}(X)$	Y em radianos
Arco-tangente	$Y = \text{ATAN}(X)$	Y em radianos
Seno hiperbólico	$Y = \text{SINH}(X)$	
Cosseno hiperbólico	$Y = \text{COSH}(X)$	
Tangente hiperbólica	$Y = \text{TANH}(X)$	

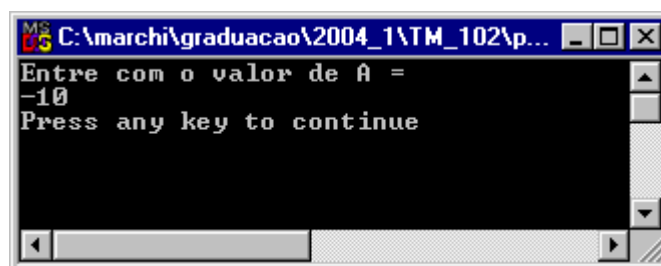


Figura 5.7 Janela DOS do programa5d.f90.

```
saida5d - Notepad
File Edit Search Help
Valor de A = -10.000000
B = Módulo de A = 10.000000
C: Logaritmo decimal de B = 1.000000
Pi = 3.141593
Cosseno de Pi = -1.000000
Cosseno hiperbólico de Pi = 11.591950
```

Figura 5.8 Arquivo saida5d.txt do programa5d.f90.

5.5 EXERCÍCIOS

Exercício 5.1

- 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 A_1
 - b) ler a diferença (número real) 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 valor (real) do 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 no arquivo chamado saida_5p1.txt 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 $A_1 = 1.3$, $D = 3.9$ e $N = 5$. Os resultados devem ser $A_N = 16.9$ e $S_N = 45.5$.

Exercício 5.2

- 1) Editar um programa-fonte em FORTRAN para executar o seguinte algoritmo (passos):
 - a) ler algumas variáveis do tipo real
 - b) aplicar as funções matemáticas da Tabela 5.5 sobre as variáveis lidas
 - c) escrever no arquivo chamado saida_5p2.txt os valores lidos e os resultados calculados com as funções juntamente com comentários para identificá-los
- 2) Compilar o programa-fonte
- 3) Gerar o programa-executável
- 4) Executar o programa. Em seguida, comparar os resultados escritos com aqueles obtidos de uma calculadora.

Exercício 5.3

- 1) Editar um programa-fonte em FORTRAN para executar o seguinte algoritmo (passos):
 - a) ler algumas variáveis do tipo real
 - b) aplicar as funções matemáticas da Tabela 5.6 sobre as variáveis lidas
 - c) escrever no arquivo chamado saida_5p3.txt os valores lidos e os resultados calculados com as funções juntamente com comentários para identificá-los
- 2) Compilar o programa-fonte
- 3) Gerar o programa-executável
- 4) Executar o programa. Em seguida, comparar os resultados escritos com aqueles obtidos de uma calculadora.