

Capítulo 5. Arquivos e funções matemáticas intrínsecas

25 Mar 2025

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 Bloco de Notas (Notepad) para mostrar o conteúdo de um arquivo durante a execução de um programa em FORTRAN
- Usar algumas funções matemáticas intrínsecas do FORTRAN
- Novos comandos do FORTRAN a usar: WRITE(número,*), OPEN, CLOSE, USE, SYSTEM

5.1 programa5a.f90

Para inicializar as atividades deste capítulo, deve-se:

- 1) Executar o programa Fortran
- 2) No Fortran, **criar um projeto** com o nome **programa05**
- 3) No Fortran, **criar e inserir** no projeto o programa-fonte **programa5a.f90**
- 4) Dentro do espaço de edição do Fortran, conforme é mostrado na Figura 5.1, **copiar** exatamente o texto em vermelho mostrado na **Tabela 5.1**.

The screenshot shows the Microsoft Developer Studio interface for a project named "programa05". The left pane displays the file structure under "programa05 files", with "programa5a.f90" selected. The right pane is the code editor containing the following Fortran code:

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

Figura 5.1 Programa5a.f90.

- 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 do primeiro asterisco no comando WRITE equivale a empregar o número 6, que é o dispositivo padrão de saída.
- 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.
- c) Se for usado um número diferente de 6, o programa não será executado normalmente.

Tabela 5.1 Programa5a.f90.

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

- 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:

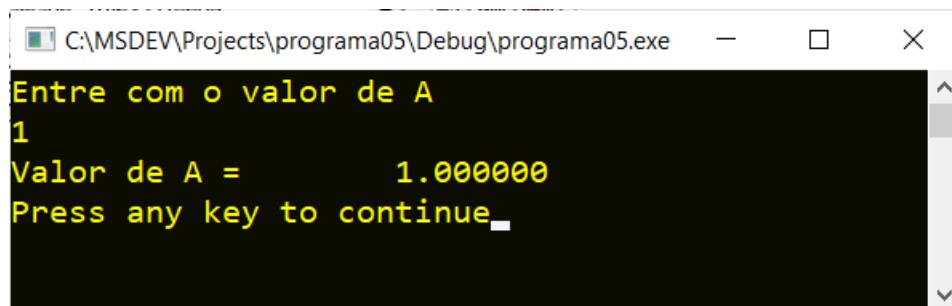


Figura 5.2 Resultado da execução do programa5a.f90.

- 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.
- b) Na segunda linha, o programa pára e fica aguardando até que seja fornecido o valor da variável A, resultado do comando **READ(*,*) A** do programa. Para que o programa continue a sua execução, é necessário **digitar o valor 1 para a variável A**, por exemplo, e, em seguida, **clicar 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,*)`.

5.2 programa5b.f90

Nesta seção será usado o mesmo projeto da seção anterior. Portanto, deve-se executar o seguinte no Fortran:

- 1) **Clicar** sobre o nome do programa-fonte
- 2) **Edit, Cut** para retirar o programa-fonte do projeto.
- 3) **Criar e inserir** no projeto o programa-fonte **programa5b.f90**
- 4) Dentro do espaço de edição do Fortran, **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 é", A  
CLOSE(7)  
  
END
```

- 5) 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 criar o arquivo chamado “nome”. A utilização deste número no comando `WRITE(número,*)` resulta na escrita de comentários ou variáveis dentro do arquivo que corresponde ao número definido no comando OPEN.
 - c) O nome do arquivo pode ser qualquer um, mas ele deve ficar entre aspas; recomenda-se usar a extensão “.txt” para o arquivo ser reconhecido pelo aplicativo Notepad (Bloco de Notas).
 - d) Quando se declara um nome de arquivo no comando OPEN, se o arquivo não existir, ele é criado dentro da pasta do projeto.

- e) 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.
- f) Todos os arquivos abertos com o comando OPEN são fechados ao final da execução do programa, existindo ou não o comando CLOSE.
- g) Dentro de um programa, para um mesmo número ou arquivo, a seqüência obrigatória dos comandos é a seguinte: OPEN, WRITE, CLOSE.
- h) Para um mesmo arquivo, deve existir apenas um comando OPEN e um CLOSE dentro do programa, e podem existir diversos comandos WRITE e outros com diversas instruções.
- i) Podem ser usados vários arquivos simultaneamente no programa.
- j) 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.
- k) Textos escritos em arquivo podem conter acentuação e símbolos, enquanto que no dispositivo padrão isto não é possível.
- l) Nada mais pode ser escrito no arquivo após o comando CLOSE no programa.
- 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 5.3, dentro da qual tem-se:

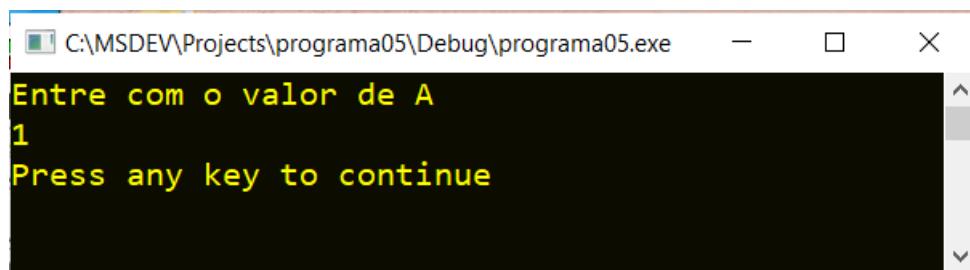


Figura 5.3 Resultado da execução do programa5b.f90 na janela DOS com A = 1.

- 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 até que seja fornecido o conteúdo da variável A, resultado do comando `READ(*,*) A` do programa. Para que o programa continue a sua execução é necessário **digitar o valor 1 para a variável A**, por exemplo, e, em seguida, **clicar na tecla enter**.
- c) Em seguida ocorrerá a criação do arquivo saída5b.txt, dentro da pasta do projeto, como resultado do comando `OPEN(7, file="saída5b.txt")` do programa, uma vez que este arquivo ainda não existe quando o programa5b.f90 for executado pela primeira vez. Este comando também declara que será usado o número 7 no programa como referência ao arquivo saída5b.txt.

- d) Conforme é mostrado na Figura 5.4, na primeira linha do arquivo saída5b.txt está escrito o comentário “Valor de A é” e o valor da variável real A como resultado do comando `WRITE(7,*)` "Valor de A é", A do programa. Para acessar o conteúdo deste arquivo, é necessário localizá-lo na pasta do projeto, com o navegador Windows Explorer, e clicar duas vezes sobre ele.

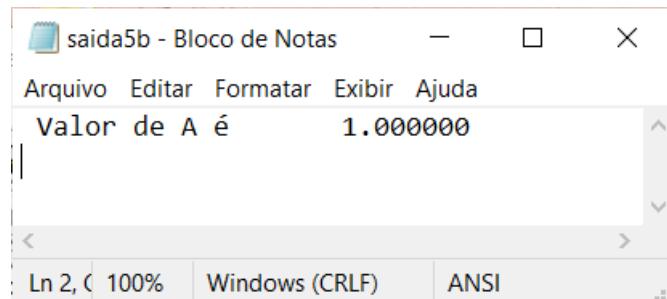


Figura 5.4 Resultado da **execução do programa5b.f90** no arquivo **saída5b.txt** com **A = 1**.

- 9) Executar novamente o programa, através de **Build, Execute**, com **A = -123**. Neste caso, como já existe o arquivo saída5b.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. **Abrir novamente o arquivo saída5b.txt** para ver o seu novo conteúdo.
- 10) No comando OPEN também é possível especificar todo o caminho ou a pasta onde o arquivo deve ser criado ou aberto. Quando o caminho não é especificado, por default, usa-se a pasta do projeto. Por exemplo:

 - a) **Trocar** a linha `OPEN(7, file="saída5b.txt")`
por `OPEN(7, file="C:\MSDEV\Projects\saída5b.txt")`
 - b) **Compilar** novamente o programa5b.f90
 - c) **Gerar** o seu executável
 - d) **Executar** o programa.
 - e) Usar o aplicativo Windows Explorer para **encontrar** e **abrir** o arquivo saída5b.txt, que agora deverá estar na pasta C:\MSDEV\Projects.

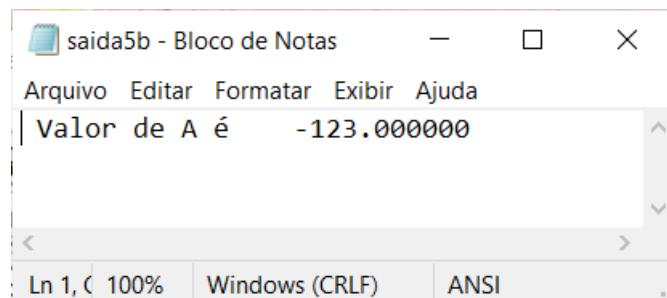


Figura 5.5 Resultado da **execução do programa5b.f90** no arquivo **saída5b.txt** com **A = -123**.

- 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

Nesta seção será usado o mesmo projeto da seção anterior. Portanto, deve-se executar o seguinte no Fortran:

- 1) **Clicar** sobre o nome do programa-fonte
- 2) **Edit, Cut** para retirar o programa-fonte do projeto
- 3) **Criar e inserir** no projeto o programa-fonte **programa5c.f90**
- 4) Dentro do espaço de edição do Fortran, **copiar** exatamente o texto em vermelho mostrado na **Tabela 5.3**.

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,*) "A = ", A
CLOSE(7)

VER = SYSTEM ("Notepad saida5c.txt")

END
```

- 5) 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” representada aqui genericamente por biblio, ou seja, biblio deve ser substituído pelo nome da biblioteca desejada. Uma “biblioteca”, no presente contexto, é uma coleção de comandos e instruções 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 na qual 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.

- 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 pastas ou arquivos; e qualquer outro comando possível de se executar na 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 onde se deseja executar o comando SYSTEM. Dentro de parênteses e aspas, deve-se incluir a sequência de comandos que se deseja executar no Windows.
- e) Com o sistema operacional Linux, deve-se trocar VER = SYSTEM por CALL SYSTEM.
- f) A linha **USE PORTLIB** declara que a biblioteca chamada PORTLIB será acrescentada ao programa.
- g) A linha **VER = SYSTEM ("Notepad saida5c.txt")** executa o comando SYSTEM para que o aplicativo BLOCO DE NOTAS (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.
- 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**, 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.
- 9) No caso do comando OPEN ser usado para especificar todo o caminho ou a pasta onde 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:\MSDEV\Projects com o comando
OPEN(7, file="C:\MSDEV\Projects\saida5c.txt"), o comando correspondente para abrir automaticamente o arquivo saida5c.txt é **VER = SYSTEM ("Notepad C:\MSDEV\Projects\saida5c.txt")**.

5.4 programa5d.f90

- Nesta seção será usado o mesmo projeto da seção anterior. Portanto, deve-se executar o seguinte no Fortran:
- 1) **Clicar** sobre o nome do programa-fonte
 - 2) **Edit, Cut** para retirar o programa-fonte do projeto
 - 3) **Criar e inserir** no projeto o programa-fonte **programa5d.f90**
 - 4) Dentro do espaço de edição do Fortran, **copiar** exatamente o texto em vermelho mostrado na **Tabela 5.4**.
 - 5) 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; isto significa que o argumento destas funções tem que ser um número

real; ele não pode ser inteiro. Havendo algum argumento que seja inteiro, basta multiplicá-lo por 1.0 para torná-lo real.

- b) Outras funções são apresentadas nas Tabelas 5.5 e 5.6.
- 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 Logarithmic Procedures.

Tabela 5.4 Programa5d.f90.

```
USE PORTLIB

REAL A, B, C, Pi, D
INTEGER VER

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

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

WRITE(8,*) "A (dato) =", 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,*) "D = cosseno de Pi =", D

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

CLOSE(8)

VER = SYSTEM ("Notepad saida5d.txt")

END
```

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

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

The screenshot shows the Microsoft Developer Studio interface with the title bar "Microsoft Developer Studio - programa05 - [programa5d.f90]". The menu bar includes File, Edit, View, Insert, Build, Tools, Window, and Help. The toolbar contains various icons for file operations like Open, Save, Print, and a search bar labeled "at this point". Below the toolbar is a dropdown menu showing "programa05 - Win32 Debug" and several status indicators. The main window displays the Fortran source code:

```

USE PORTLIB

REAL A, B, C, Pi, D
INTEGER VER

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

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

WRITE(8,*) "A (dado) =", 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,*) "D = cosseno de Pi =", D

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

CLOSE(8)

VER = SYSTEM ("Notepad saida5d.txt")

END

```

At the bottom, there are tabs for "FileView" and "InfoView", and a status bar showing "Ready", "Ln 33, Col 1", and "REC COL".

Figura 5.6 Programa5d.f90.

8) 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 (REAL)”, resultado do comando

`WRITE(*,*) "Entre com o valor de A (REAL)"` do programa.

- b) Na segunda linha, o programa pára e fica aguardando até que seja fornecido o valor da variável A, resultado do comando `READ(*,*) A` do programa. Para que o programa continue a 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 saída5d.txt, dentro da pasta do projeto, como resultado do comando `OPEN(8, file="saída5d.txt")` do programa, uma vez que este arquivo ainda não existe quando o programa5d.f90 for executado pela primeira vez. Este comando também declara que será usado o número 8 no programa como referência ao arquivo saída5d.txt.
- d) Os resultados das funções aplicadas no programa5d.f90 são mostrados na Figura 5.8.

Tabela 5.5 Algumas funções matemáticas intrínsecas do FORTRAN.

Função	Comando	Observação
Módulo ou valor absoluto	$Y = ABS(X)$	
Raiz quadrada	$Y = SQRT(X)$	
Exponencial	$Y = EXP(X)$	
Logaritmo natural	$Y = LOG(X)$	
Logaritmo decimal	$Y = LOG10(X)$	
Parte inteira de número real	$Y = INT(X)$	$Y = $ parte inteira de X
Resto da divisão	$Y = MOD(X,Z)$	$Y = X - INT(X/Z)*Z;$ se $Y = 0$, a divisão é exata; se $Y \neq 0$, há resto diferente de 0 na divisão
Sinal	$Y = SIGN(X,Z)$	$Y = $ valor absoluto de X vezes o sinal de Z
Mínimo	$Y = MIN(X,Z,W)$	$Y = $ valor mínimo entre X, Z e W; o argumento pode conter 2 ou mais variáveis
Máximo	$Y = MAX(X,Z,W)$	$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	$Y = SIN(X)$	X em radianos
Seno	$Y = SIND(X)$	X em graus
Cosseno	$Y = COSD(X)$	X em graus
Tangente	$Y = TAND(X)$	X em graus
Arco-seno	$Y = ASIND(X)$	Y em graus
Arco-cosseno	$Y = ACOSD(X)$	Y em graus
Arco-tangente	$Y = ATAND(X)$	Y em graus
Seno hiperbólico	$Y = SINH(X)$	
Cosseno hiperbólico	$Y = COSH(X)$	
Tangente hiperbólica	$Y = TANH(X)$	

- 9) **Comparar** cada resultado mostrado na Figura 5.8 com o resultado obtido de uma calculadora.
- 10) **Executar** novamente o programa com outro valor para a variável A. **Comparar** cada novo resultado com o obtido de uma calculadora.

11) Os programas-executáveis também podem ser executados, rodados ou postos para funcionar fora do aplicativo Fortran ou mesmo em algum computador que não tenha o aplicativo Fortran. Para testar isso, primeiro, **fechar o aplicativo Fortran**. Em seguida, utilizando o aplicativo Windows Explorer, **copiar** o programa-executável chamado **programa05.exe ou programa05.aplicativo**, que está dentro da pasta DEBUG, dentro da pasta do projeto, **para a pasta C:\MSDEV\Projects**. Para executar o programa, basta **clicar duas vezes sobre o arquivo copiado**. Como resultado, surgirá o arquivo saída5d.txt. A pasta usada, C:\MSDEV\Projects, é apenas um exemplo. O programa-executável pode ser copiado para qualquer outra pasta ou computador e ser executado lá.

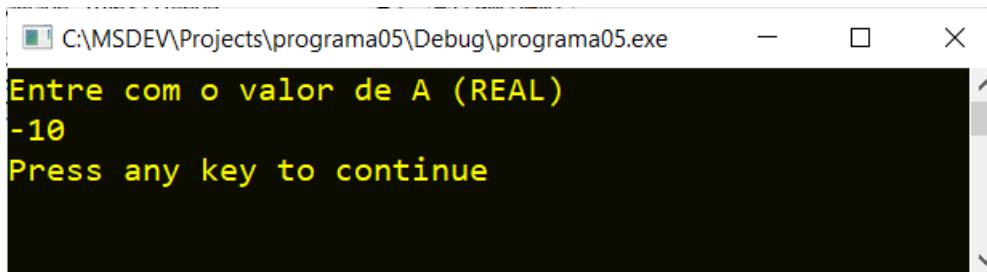


Figura 5.7 Janela DOS do programa5d.f90.

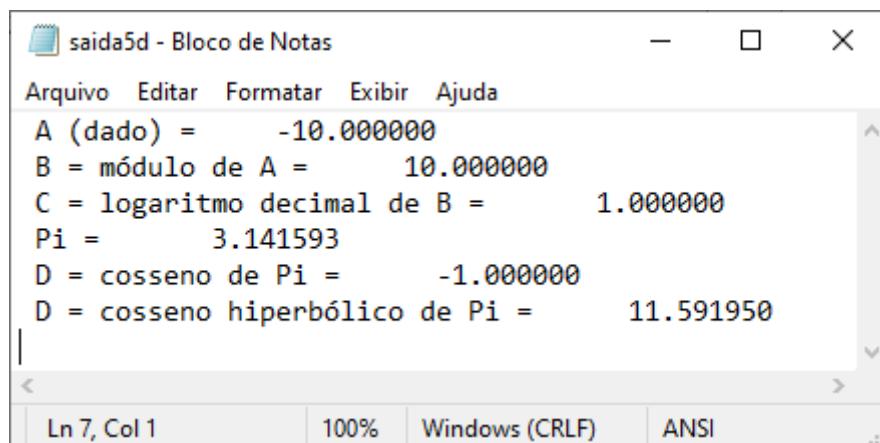


Figura 5.8 Arquivo saída5d.txt do programa5d.f90.

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 A1
 - b) ler a diferença (número real) entre dois termos subsequentes 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 AN, com $AN = A1 + (N-1)*D$
 - e) calcular a soma de todos os termos da P.A., denotada por SN, com a seguinte expressão:
$$SN = (A1+AN)*N/2$$
 - f) escrever, no arquivo chamado saída_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 $A1 = 1.3$, $D = 3.9$ e $N = 5$. Os resultados devem ser $AN = 16.9$ e $SN = 45.5$.

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 algumas funções matemáticas da Tabela 5.5 sobre as variáveis lidas
 - c) escrever, no arquivo chamado saída_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 algumas funções matemáticas da Tabela 5.6 sobre as variáveis lidas
 - c) escrever, no arquivo chamado saída_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.

Exercício 5.4

Editar um programa-fonte em FORTRAN para executar o seguinte algoritmo (passos):

- 1) Declarar as variáveis G, R, TG e TR como reais
- 2) Ler o valor de G, que representará um ângulo em graus
- 3) Transformar G em radianos, atribuindo o resultado à variável R
- 4) Calcular a tangente de G e atribuir o resultado a TG
- 5) Calcular a tangente de R e atribuir o resultado a TR
- 6) Escrever, no arquivo chamado saída_5p4.txt, os valores lidos e os resultados calculados juntamente com comentários para identificá-los
- 7) Executar o programa para $G = 45^\circ$. Os resultados esperados são $TG = TR = 1.000000$

Exercício 5.5

Editar um programa-fonte em FORTRAN para executar o seguinte algoritmo (passos):

- 1) Criar um projeto com o nome **Cap5** e inserir nele o programa-fonte **Cap5.f90**
- 2) Ler os valores de I, K e L.
- 3) Escrever o nome completo do aluno no arquivo **Cap5.TXT**
- 4) Calcular:

$$J = \text{raiz quadrada de } K$$

$$M = \text{exponencial de } L$$

$$N = \text{logaritmo natural de } I$$

- 5) Escrever no arquivo Cap5.TXT os resultados de J, M e N junto com seus nomes.
- 6) Abrir o arquivo Cap5.TXT com o aplicativo Bloco de Notas (Notepad)
- 7) Executar o programa com $I = 4$, $K = 2$ e $L = 3$. Os resultados esperados são:

$$J = 1.414214$$

$$M = 20.085540$$

$$N = 1.386294$$

Exercício 5.6

Editar um programa-fonte em FORTRAN para executar o seguinte algoritmo (passos):

- 1) Criar um projeto com o nome **Cap5b** e inserir nele o programa-fonte **Cap5b.f90**
- 2) Ler os valores de I e L.
- 3) Escrever o nome completo do aluno no arquivo **Cap5b.TXT**
- 4) Incluindo o nome de cada variável calculada, escrever no arquivo Cap5b.TXT os resultados de:

$$K = \text{logaritmo decimal de } I$$

$$M = \text{tangente de } L$$

- 5) Abrir o arquivo Cap5b.TXT com o aplicativo Bloco de Notas (Notepad)
- 6) Executar o programa com $I = 1000$ e $L = 45$ graus. Os resultados esperados são:

$$K = 3.0000000$$

$$M = 1.0000000$$

Exercício 5.7

Editar um programa-fonte em FORTRAN para executar o seguinte algoritmo (passos):

- 1) Definir Y como uma variável do tipo inteiro; e K e L do tipo real.
- 2) Ler os valores de Y, K e L.
- 3) Incluindo comentários para identificar cada valor, escrever no arquivo Resulta5.TXT, os resultados de:
 - a) Raiz quadrada de K
 - b) Exponencial de L
 - c) Logaritmo natural de Y
 - d) Mínimo entre Y, K e L
- 4) Abrir o arquivo Resulta5.TXT com o aplicativo Notepad

Resultados para Y = 4, K = 2 e L = 3:

- a) 1.414214 b) 20.085540 c) 1.386294 d) 2.000000