



UNIVERSIDADE FEDERAL DO PARANÁ

CURSO DE ENGENHARIA MECÂNICA

**Disciplina: TM-102 INFORMÁTICA / PROGRAMAÇÃO****Semestre: 2003/2**

## 9ª aula

### INSTRUÇÕES

- Siga o roteiro abaixo.
- Qualquer dúvida, chame o professor.

### 9.1 OBJETIVOS DA AULA

- 1) MATRIZES
- 2) CICLOS DUPLOS

### 9.2 CRIANDO UM NOVO PROJETO DO TIPO QUICKWIN APPLICATION

- 1) Na “janela principal” do Fortran execute: File, New
- 2) Na janela “New”: selecione “Project Workspace” e clique sobre o botão “OK”
- 3) Na janela “New Project Workspace”, execute o seguinte:
  - a) Type: selecione “QuickWin Application”
  - b) Location: use o botão BROWSE para indicar o seu diretório no “drive” Server1, diretório (pasta) “alunos\_2003\_2”.
  - c) Name: indique o nome do projeto. Use “programa09”. O software criará um diretório com este nome, dentro do diretório indicado no passo anterior.
  - d) Clique em “Create”
  - e) No lado esquerdo da “janela principal” deverá aparecer o nome do projeto criado

### 9.3 INSERINDO UM ARQUIVO DENTRO DO PROJETO

- 1) Na “janela principal” do Fortran, execute: Insert, Files into Project
- 2) Na janela “Insert Files into Project”:
  - a) File Name: entre com o nome do arquivo a ser criado. Use “programa9a.f90”. Clique em “OK”
  - b) Na janela “Microsoft Developer Studio”, clique em YES
  - c) No lado esquerdo da “janela principal”, clique sobre o sinal + ao lado do nome do projeto; deverá aparecer o nome do arquivo que foi inserido.
  - d) Clique duas vezes sobre o nome do arquivo que foi inserido
  - e) Na janela “Microsoft Developer Studio”, clique em YES
  - f) O arquivo inserido está pronto para ser editado no lado direito da “janela principal”

## 9.4 PROGRAMA9a.f90

- 1) Dentro do espaço de edição de programas do Fortran, escreva o programa abaixo (para o programa funcionar, basta digitar as linhas em vermelho, as demais são comentários que você deve ler e entender):

```
! -----  
! programa9a.f90  
! -----  
! Objetivo: definir uma MATRIZ de dimensão alocável, com elementos do  
!           tipo inteiro, ler e escrever seus componentes em arquivos  
! -----  
! Parte do programa onde são incluídas bibliotecas (módulos) do Fortran  
use msflib  
! Incluí no programa a biblioteca do Fortran que permite usar o comando  
! "systemqq"  
! -----  
! Parte do programa onde são definidas as variáveis do programa  
implicit none  
! Ao se usar este comando, todas as variáveis do programa têm que ser definidas  
! explicitamente com os comandos real, character, integer e logical. Assim,  
! evita-se erros de lógica no programa, captados ao se fazer a compilação.  
logical dos  
! Declaração de variável do tipo "lógica" para usar o comando "systemqq"  
integer linhas  
! Declaração da variável "linhas" como sendo do tipo inteiro. Seu valor  
! corresponde à quantidade de linhas da matriz  
integer linha  
! Declaração da variável "linha" como sendo do tipo inteiro. Seu valor  
! corresponde à uma linha específica da matriz  
integer colunas  
! Declaração da variável "colunas" como sendo do tipo inteiro. Seu valor  
! corresponde à quantidade de colunas da matriz  
integer coluna  
! Declaração da variável "coluna" como sendo do tipo inteiro. Seu valor  
! corresponde à uma coluna específica da matriz  
integer,dimension(:,),allocatable :: matriz_a  
! Declaração da variável chamada "matriz_a" com as seguintes características:  
! (1) comando INTEGER a define como sendo do tipo inteiro;  
! (2) comando DIMENSION(:,) a define com dimensão alocável ou aberta, conforme  
!     o comando ALLOCATABLE, isto é, suas dimensões serão definidas dentro do  
!     programa em função de alguma outra variável;  
! (3) a matriz é bidimensional por ter sido definida com duas vezes  
!     dois pontos, separados por vírgula, no comando DIMENSION(:,)  
! -----  
! Parte do programa onde são lidos os dados  
dos = systemqq('Notepad dados9a.txt')  
! Mostra o conteúdo do arquivo "dados9a.txt" com o programa Notepad  
open(12,file='dados9a.txt')  
! Define o número "12" como uma referência ao arquivo "dados9a.txt"
```

```
read(12,*) linhas
```

```
read(12,*) colunas
```

```
! Nas duas linhas acima, utiliza-se o comando READ para ler as variáveis  
! "linhas" e "colunas" do arquivo 12, que corresponde ao arquivo  
! "dados9a.txt", conforme definido no comando OPEN acima.  
! Como há um asterisco, o conteúdo de cada variável é lido sem qualquer formato.  
! Como há um comando READ para cada variável, é lida apenas uma variável por  
! linha no arquivo "dados9a.txt".
```

```
allocate ( matriz_a ( linhas, colunas ) )
```

```
! Com este comando ALLOCATE define-se quantos elementos terá a variável  
! "matriz_a". Note que esta definição ocorre em função das variáveis "linhas" e  
! "colunas" que são dados do programa. Portanto, a alocação de memória para a  
! variável "matriz_a" não é pré-definida, ela ocorre em função de dados do  
! programa.
```

```
do linha = 1, linhas
```

```
! Este ciclo DO inicia com o valor 1 e termina com o valor da variável "linhas",  
! tendo como passo o valor unitário (por não ter sido especificado).  
! É um ciclo do tipo progressivo, definido por variáveis inteiras.  
! A variável "linha" começará com o valor 1. Quando a execução do programa  
! atingir o comando END DO, haverá um retorno ao comando DO e o valor da  
! variável "linha" passará a ser a soma da variável "linha" com o "passo". Se o  
! novo valor da variável "linha" for menor ou igual à variável "linhas", o  
! programa voltará a executar o ciclo DO a END DO. Caso contrário, a execução do  
! programa prosseguirá a partir do comando END DO.
```

```
do coluna = 1, colunas
```

```
! Início de um ciclo DO, dentro de outro ciclo DO, que varia de 1 a "colunas"
```

```
read(12,*) matriz_a(linha,coluna)
```

```
! Este comando READ é usado para ler cada elemento da variável "matriz_a"  
! do arquivo 12, que corresponde ao arquivo "dados9a.txt", conforme  
! definido no comando OPEN acima.  
! Como há um asterisco, o valor de cada elemento da variável "matriz_a"  
! é lido sem qualquer formato.  
! Note que só há um comando READ. Mas ele está dentro de um ciclo externo,  
! que passa por todas as "linhas" da "matriz_a", e dentro de um ciclo  
! interno que passa por todas as "colunas" da "matriz_a". Portanto, neste  
! caso, cada elemento da "matriz_a" deverá estar numa linha diferente do  
! arquivo "dados9a.txt".
```

```
end do
```

```
! Encerra o ciclo DO interno
```

```
end do
```

```
! Encerra o ciclo DO externo
```

```
close(12)
```

```
! O comando CLOSE encerra o uso do arquivo 12
```

```
! -----
```

```
! Parte do programa onde a "matriz_a" é escrita de duas formas num arquivo
```

```
open(10, file='resultados9a.txt')
```

```
! Define o número "10" como uma referência ao arquivo "resultados9a.txt"
```

```
write(10,21)
```

```
! Escreve um comentário no arquivo 10 com o formato especificado pelo número 21
```

```
write(10,*) matriz_a
```

```
! Escreve a "matriz_a" no arquivo 10 sem qualquer formato
```

```
write(10,31)
```

```
! Escreve um comentário no arquivo 10 com o formato especificado pelo número 31
```

```

do linha = 1, linhas

  do coluna = 1, colunas

    write(10,32) linha, coluna, matriz_a(linha,coluna)
    ! escreve no arquivo 10, com o formato especificado pelo número 32, os
    ! valores das variáveis "linha", "coluna" e cada elemento da "matriz_a"

  end do

end do

close(10)
! O comando CLOSE encerra o uso do arquivo 10

dos = systemqq('Notepad resultados9a.txt')
! Mostra o conteúdo do arquivo "resultados9a.txt" com o programa Notepad

! -----

! Parte do programa onde são definidos os formatos utilizados

21 format( 1/, 'Escrita da "matriz_a" sem formato', 1/ )

31 format( 1/, 'Escrita da "matriz_a" na mesma seqüência dos dados', 1/ )

32 format( 3x, 'matriz_a(', i1, ', ', i1, ') = ', i5)

end
! O comando END encerra o programa

```

- 2) Depois de digitar o programa, grave-o da seguinte forma: File, Save
- 3) Compile o programa executando: Build, Compile
- 4) Gere o executável do programa com Build, Build
- 5) Antes de executar este programa, você precisa criar o arquivo “dados9a.txt”, inserindo nele o número de linhas e colunas e todos os elementos da matriz\_a. Para fazer isso, execute o seguinte:
  - a) Start, Programs, Accessories, Notepad
  - b) Dentro do espaço de edição do Notepad, edite por exemplo:
 

```

3 número de linhas da "matriz_a"
2 número de colunas da "matriz_a"
1 elemento da linha 1 coluna 1
4 elemento da linha 1 coluna 2
2 elemento da linha 2 coluna 1
5 elemento da linha 2 coluna 2
3 elemento da linha 3 coluna 1
6 elemento da linha 3 coluna 2

```
  - c) File, Save
  - d) Na janela “Save As”, em “File Name”, coloque “dados9a.txt”; em “Save in:”, selecione o diretório do seu projeto. Em seguida, clique sobre o botão “Save”. Finalmente, selecione File, Exit.
- 6) Note que os dados acima correspondem à seguinte matriz:
 

1	4
2	5
3	6
- 7) Execute o programa com Build, Execute. O resultado deverá ser:
  - a) A abertura de uma janela do DOS.
  - b) A abertura de uma janela do WINDOWS.
  - c) A abertura do arquivo “dados9a.txt” com o programa Notepad. Se o arquivo não existir no diretório do projeto, ocorrerá um erro. Se você quiser, altere o conteúdo das variáveis no arquivo “dados9a.txt”. Depois, grave o novo valor com File, Save. Finalmente, feche este arquivo com File, Exit.

- d) Em seguida, ocorrerá a abertura do arquivo “**resultados9a.txt**” com o programa Notepad. O conteúdo deverá ser:

**Escrita da "matriz\_a" sem formato**

**1                    2                    3                    4                    5                    6**

**Escrita da "matriz\_a" na mesma seqüência dos dados**

```
matriz_a(1,1) = 1
matriz_a(1,2) = 4
matriz_a(2,1) = 2
matriz_a(2,2) = 5
matriz_a(3,1) = 3
matriz_a(3,2) = 6
```

- e) Até entender, compare o conteúdo de cada linha do arquivo “**resultados9a.txt**” com os comandos do programa9a.f90 e seus respectivos formatos de edição.
- f) Note que quando não se usa formato, a ordem de escrita dos elementos da matriz\_a é por coluna, e não por linha, que é o formato empregado na leitura e escrita dos dados.
- g) Feche o arquivo “**resultados9a.txt**” com File, Exit.
- h) Em seguida, clique na opção “Yes” para encerrar a execução do programa.
- 8) Execute novamente o programa com Build, Execute. Altere os valores dos elementos, o número de colunas e de linhas.

## **9.5 PROGRAMA9b.f90**

- 1) Clique dentro do espaço de edição do seu “**programa9a.f90**”. Depois execute: File, Close. Clique sobre o nome do “**programa9a.f90**”, no lado esquerdo da janela principal do Fortran. Em seguida, execute: Edit, Cut. Com isso, o “**programa9a.f90**” será eliminado do seu projeto, mas o arquivo continuará a existir dentro do diretório do seu projeto, podendo ser incluído nele quando desejado.
- 2) Seguindo os passos da seção 9.3 deste tutorial, insira no projeto “**programa09**” o arquivo “**programa9b.f90**”.
- 3) Dentro do espaço de edição de programas do Fortran, escreva o programa abaixo (para o programa funcionar, basta digitar as linhas em vermelho, as demais são comentários que você deve ler e entender):

```
! -----
! programa9b.f90
! -----
! Objetivo: definir MATRIZES de dimensão alocável, com elementos do
!           tipo real, e realizar operações matemáticas com elas
! -----
! Parte do programa onde são incluídas bibliotecas (módulos) do Fortran
use msflib
! Inclui no programa a biblioteca do Fortran que permite usar o comando
! "systemqq"
! -----
! Parte do programa onde são definidas as variáveis do programa
implicit none
! Ao se usar este comando, todas as variáveis do programa têm que ser definidas
! explicitamente com os comandos real, character, integer e logical. Assim,
```

! evita-se erros de lógica no programa, captados ao se fazer a compilação.

**logical dos**

! Declaração de variável do tipo "lógica" para usar o comando "systemqq"

**integer linhas**

! Declaração da variável "linhas" como sendo do tipo inteiro. Seu valor  
! corresponde à quantidade de linhas das matrizes

**integer linha**

! Declaração da variável "linha" como sendo do tipo inteiro. Seu valor  
! corresponde à uma linha específica da matriz

**integer colunas**

! Declaração da variável "colunas" como sendo do tipo inteiro. Seu valor  
! corresponde à quantidade de colunas das matrizes

**integer coluna**

! Declaração da variável "coluna" como sendo do tipo inteiro. Seu valor  
! corresponde à uma coluna específica da matriz

**integer constante**

! Declaração da variável "constante" como sendo do tipo inteiro.

**real\*8,dimension(:,:),allocatable :: matriz\_a, matriz\_b, matriz\_c**

! Declaração das variáveis chamadas "matriz\_a", "matriz\_b" e "matriz\_c" com as  
! seguintes características:

! (1) comando REAL\*8 as define como sendo do tipo real dupla;  
! (2) comando DIMENSION(:,:) as define com dimensão alocável ou aberta, conforme  
! o comando ALLOCATABLE, isto é, suas dimensões serão definidas dentro do  
! programa em função de alguma outra variável;  
! (3) as matrizes são bidimensionais por terem sido definidas com duas vezes  
! dois pontos, separados por vírgula, no comando DIMENSION(:,:)

! -----

! Parte do programa onde são lidos os dados

**dos = systemqq('Notepad dados9b.txt')**

! Mostra o conteúdo do arquivo "dados9b.txt" com o programa Notepad

**open(12,file='dados9b.txt')**

! Define o número "12" como uma referência ao arquivo "dados9b.txt"

**read(12,\*) constante**

**read(12,\*) linhas**

**read(12,\*) colunas**

! Nas três linhas acima, utiliza-se o comando READ para ler as variáveis  
! "constante", "linhas" e "colunas" do arquivo 12, que corresponde ao arquivo  
! "dados9b.txt", conforme definido no comando OPEN acima.  
! Como há um asterisco, o conteúdo de cada variável é lido sem qualquer formato.  
! Como há um comando READ para cada variável, é lida apenas uma variável por  
! linha no arquivo "dados9b.txt".

**allocate ( matriz\_a ( linhas, colunas ), matriz\_b ( linhas, colunas ), &  
          matriz\_c ( linhas, colunas ) )**

! Com este comando ALLOCATE define-se quantos elementos terão as variáveis  
! "matriz\_a", "matriz\_b" e "matriz\_c". Note que esta definição ocorre em função  
! das variáveis "linhas" e "colunas" que são dados do programa. Portanto, a  
! alocação de memória para as variáveis "matriz\_a", "matriz\_b" e "matriz\_c" não  
! é pré-definida, ela ocorre em função de dados do programa.

**do linha = 1, linhas**

! Este ciclo DO inicia com o valor 1 e termina com o valor da variável "linhas",  
! tendo como passo o valor unitário (por não ter sido especificado).  
! É um ciclo do tipo progressivo, definido por variáveis inteiras.

```
! A variável "linha" começará com o valor 1. Quando a execução do programa
! atingir o comando END DO, haverá um retorno ao comando DO e o valor da
! variável "linha" passará a ser a soma da variável "linha" com o "passo". Se o
! novo valor da variável "linha" for menor ou igual à variável "linhas", o
! programa voltará a executar o ciclo DO a END DO. Caso contrário, a execução do
! programa prosseguirá a partir do comando END DO.
```

```
read(12,*) (matriz_a(linha,coluna), coluna=1,colunas)
```

```
! Este comando READ é usado para ler cada "linha" da variável "matriz_a" do
! arquivo 12, que corresponde ao arquivo "dados9b.txt", conforme definido no
! comando OPEN acima.
```

```
! Como há um asterisco, o conteúdo de cada variável é lido sem qualquer
! formato.
```

```
! Neste comando READ, para que seja possível ler todos os elementos de uma
! mesma linha da "matriz_a", está sendo usado o que se chama de COMANDO
```

```
! DO IMPLÍCITO OU ANINHADO. Note que só há um comando READ. Mas ele está
! dentro de um ciclo, que passa por todas as "linhas" da "matriz_a", e na
! própria linha do comando READ há um outro ciclo, que passa por todas as
! "colunas" da "matriz_a".
```

```
end do
```

```
! Encerra o ciclo DO externo
```

```
close(12)
```

```
! O comando CLOSE encerra o uso do arquivo 12
```

```
! -----
```

```
! Parte do programa onde é gerada a "matriz_b" através da multiplicação de uma
! constante pela "matriz_a"
```

```
do linha = 1, linhas
```

```
! Início de um ciclo DO que varia de 1 a "linhas"
```

```
do coluna = 1, colunas
```

```
! Início de um ciclo DO, dentro de outro ciclo DO, que varia de 1 a "colunas"
```

```
matriz_b(linha,coluna) = constante * matriz_a(linha,coluna)
```

```
! Cada elemento da "matriz_b" é obtido ao se multiplicar a variável
! "constante" por cada elemento da "matriz_a"
```

```
end do
```

```
! Encerra o ciclo DO interno
```

```
end do
```

```
! Encerra o ciclo DO externo
```

```
! -----
```

```
! Parte do programa onde é gerada a "matriz_c" através da soma da "matriz_a" com
! a "matriz_b"
```

```
do linha = 1, linhas
```

```
! Início de um ciclo DO que varia de 1 a "linhas"
```

```
do coluna = 1, colunas
```

```
! Início de um ciclo DO, dentro de outro ciclo DO, que varia de 1 a "colunas"
```

```
matriz_c(linha,coluna) = matriz_a(linha,coluna) + matriz_b(linha,coluna)
```

```
! Cada elemento da "matriz_c" é obtido ao se somar cada elemento da
! "matriz_a" com cada elemento da "matriz_b"
```

```
end do
```

```
! Encerra o ciclo DO interno
```

```
end do
```

```
! Encerra o ciclo DO externo
```

```
! -----
```

! Parte do programa onde são escritos os resultados num arquivo

```
open(10,file='resultados9b.txt')
```

! Define o número "10" como uma referência ao arquivo "resultados9b.txt"

```
write(10,31)
```

! Escreve um comentário no arquivo 10 com o formato especificado pelo número 31

```
do linha = 1, linhas
```

```
  write(10,32) (matriz_a(linha,coluna), coluna=1,colunas)
```

! escreve no arquivo 10, com o formato especificado pelo número 32, os  
! valores dos elementos da "matriz\_a"

```
end do
```

```
write(10,41)
```

! Escreve um comentário no arquivo 10 com o formato especificado pelo número 41

```
do linha = 1, linhas
```

```
  write(10,32) (matriz_b(linha,coluna), coluna=1,colunas)
```

! escreve no arquivo 10, com o formato especificado pelo número 32, os  
! valores dos elementos da "matriz\_b"

```
end do
```

```
write(10,51)
```

! Escreve um comentário no arquivo 10 com o formato especificado pelo número 51

```
do linha = 1, linhas
```

```
  write(10,32) (matriz_c(linha,coluna), coluna=1,colunas)
```

! escreve no arquivo 10, com o formato especificado pelo número 32, os  
! valores dos elementos da "matriz\_c"

```
end do
```

```
close(10)
```

! O comando CLOSE encerra o uso do arquivo 10

```
dos = systemqq('Notepad resultados9b.txt')
```

! Mostra o conteúdo do arquivo "resultados9b.txt" com o programa Notepad

```
! -----
```

! Parte do programa onde são definidos os formatos utilizados

```
31 format(3x, '*** MATRIZ A (dados) ***')
```

```
41 format(1/, 3x, '*** MATRIZ B (constante * matriz A) ***')
```

```
51 format(1/, 3x, '*** MATRIZ C (matriz A + matriz B) ***')
```

```
32 format(10(3x,1pe10.3))
```

```
end
```

! O comando END encerra o programa

4) Depois de digitar o programa, grave-o da seguinte forma: File, Save

5) Compile o programa executando: Build, Compile

6) Gere o executável do programa com Build, Build

7) Antes de executar este programa, você precisa criar o arquivo "dados9b.txt", inserindo nele uma constante, o número de linhas e colunas e todos os elementos da matriz\_a. Para fazer isso, execute o seguinte:

a) Start, Programs, Accessories, Notepad

b) Dentro do espaço de edição do Notepad, edite por exemplo:

```
5      constante (inteiro)
3      número de linhas da "matriz_a" (inteiro)
2      número de colunas da "matriz_a" (inteiro)
```



```

1 4 todos os elementos da linha 1 da "matriz_a"
2 5 todos os elementos da linha 2 da "matriz_a"
3 6 todos os elementos da linha 3 da "matriz_a"

```

c) File, Save

d) Na janela “Save As”, em “File Name”, coloque “dados9b.txt”; em “Save in:”, selecione o diretório do seu projeto. Em seguida, clique sobre o botão “Save”. Finalmente, selecione File, Exit.

8) Execute o programa com Build, Execute. O resultado deverá ser:

a) A abertura de uma janela do DOS.

b) A abertura de uma janela do WINDOWS.

c) A abertura do arquivo “dados9b.txt” com o programa Notepad. Se o arquivo não existir no diretório do projeto, ocorrerá um erro. Se você quiser, altere o conteúdo das variáveis no arquivo “dados9b.txt”. Depois, grave o novo valor com File, Save. Finalmente, feche este arquivo com File, Exit.

d) Em seguida, ocorrerá a abertura do arquivo “resultados9b.txt” com o programa Notepad. O conteúdo deverá ser:

```

*** MATRIZ A (dados) ***
1.000E+00 4.000E+00
2.000E+00 5.000E+00
3.000E+00 6.000E+00

*** MATRIZ B (constante * matriz A) ***
5.000E+00 2.000E+01
1.000E+01 2.500E+01
1.500E+01 3.000E+01

*** MATRIZ C (matriz A + matriz B) ***
6.000E+00 2.400E+01
1.200E+01 3.000E+01
1.800E+01 3.600E+01

```

e) Até entender, compare o conteúdo de cada linha do arquivo “resultados9b.txt” com os comandos do programa9b.f90 e seus respectivos formatos de edição.

f) Feche o arquivo “resultados9b.txt” com File, Exit.

g) Em seguida, clique na opção “Yes” para encerrar a execução do programa.

9) Execute novamente o programa com Build, Execute. Altere os valores da constante, dos elementos, o número de colunas e de linhas.

10) Conforme visto no programa9a.f90 e no programa9b.f90, os elementos das matrizes podem assumir valores inteiros ou reais. Foram vistos exemplos apenas de matrizes bidimensionais, isto é, cujos elementos são localizados com apenas dois índices (linha e coluna). Entretanto, as matrizes, em Fortran, podem ter até 7 dimensões, e os índices também podem ser negativos.

## **9.6 ENCERRANDO A AULA**

1) Dentro da “janela principal” do Fortran, feche o seu projeto executando o seguinte: File, Close Workspace

2) Na janela “Microsoft Developer Studio” clique sobre o botão “Yes”.

3) Para fechar o programa Fortran, basta executar: File, Exit

4) Para fechar o Windows, execute: Start, Shut Down

5) Na janela “Shut Down Windows”, escolha a opção “Close all programs and log on as a different user?”. Depois clique no botão “Yes”.

6) Depois que o computador fechar o Windows, por favor, apenas desligue o monitor do computador.

## LISTA DE EXERCÍCIOS DA 9ª AULA

### Exercício 9.1

- 1) Implemente um programa em Fortran para calcular a matriz resultante do produto de duas matrizes bidimensionais. Utilize o seguinte algoritmo:
  - a) Ler do arquivo “dados9.txt”:
    - linhas\_a = número de linhas da matriz\_a;
    - colunas\_a = número de colunas da matriz\_a;
    - colunas\_b = número de colunas da matriz\_b;
    - valores dos elementos da matriz\_a;
    - valores dos elementos da matriz\_b;
  - b) Calcular o valor de cada elemento da matriz\_c que resulta do produto das matrizes a e b.
  - c) Escrever no arquivo “resultados9.txt” a matriz\_c resultante.
- 2) Compile o programa
- 3) Gere o executável do programa
- 4) Execute o programa
- 5) Entregue na próxima aula:
  - Uma impressão do seu programa fonte em Fortran;
  - Uma impressão do arquivo “dados9.txt”; e
  - Uma impressão do arquivo “resultados9.txt”.