



UNIVERSIDADE FEDERAL DO PARANÁ

CURSO DE ENGENHARIA MECÂNICA

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

7ª aula

INSTRUÇÕES

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

7.1 OBJETIVO DA AULA

1) VETORES

7.2 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 “programa07”. 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

7.3 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 “programa7a.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”

7.4 PROGRAMA7a.f90

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

```
! -----  
! Objetivo: definir um vetor com dimensão pré-definida, com elementos do  
!           tipo inteiro, e realizar operações com ele  
! -----  
!  
! Parte do programa onde são incluídas bibliotecas (módulos) do Fortran90  
  
use msflib  
! Inclui no programa a biblioteca do Fortran que permite usar os comandos  
! "systemqq" e "setexitqq"  
  
use msfwin  
! Inclui no programa a biblioteca do Fortran que permite usar o  
! comando "setwindowtext"  
!  
! -----  
!  
! 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"  
  
logical janela  
! Declaração de variável do tipo "lógica" para usar o comando "setwindowtext"  
  
integer*4 fecha  
! Declaração de variável do tipo "inteiro" para usar o comando "setexitqq"  
  
integer,dimension(5) :: vetor  
! Declaração da variável chamada "vetor" como do tipo inteiro com  
! cinco componentes, conforme especificado pelo comando "dimension(5)"  
  
integer elemento  
! Número da componente ou elemento do vetor  
  
integer soma  
! soma dos valores de cada componente do vetor  
  
integer maximo  
! valor máximo do vetor  
  
integer minimo  
! valor mínimo do vetor  
  
integer local_maximo(1)  
! número da componente ou elemento do vetor que tem o valor máximo do vetor  
  
integer local_minimo(1)  
! número da componente ou elemento do vetor que tem o valor mínimo do vetor  
!  
! -----  
  
janela = setwindowtext(gethwnqq(qwin$framewindow), &  
"Programa7a.f90 - UM VETOR COM 5 ELEMENTOS"C)
```

```
! Esta instrução define o nome da janela do programa.
! O nome deve ficar entre aspas, conforme o exemplo acima.

! -----

! Parte do programa onde são lidos os dados

dos = systemqq('Notepad dados7a.txt')
! Mostra o conteúdo do arquivo "dados7a.txt" com o programa Notepad

open(7, file='dados7a.txt')
! Define o número "7" como uma referência ao arquivo "dados7a.txt"

! Um CICLO na linguagem Fortran é definido por uma linha que inicia com o
! comando DO e outra linha que contém o comando END DO.

! Todas as linhas que estiverem entre os comandos DO e END DO serão repetidas
! de acordo com o que se define na linha do comando DO. Esta linha é definida
! por quatro variáveis:
! (1) um valor inicial do ciclo (no exemplo abaixo, usa-se o valor 1);
! (2) um valor final do ciclo (no exemplo abaixo, usa-se o valor 5);
! (3) um passo para alterar a variável "contador" do ciclo (no exemplo abaixo,
!     não foi especificado o valor do passo nem há uma variável para isso);
!     quando não se define o valor do passo, ele vale 1;
! (4) um contador (no exemplo abaixo, usa-se a variável "elemento" como contador).
! Estas quatro variáveis podem ser do tipo inteiro (como neste exemplo) ou do
! tipo real.

! O ciclo pode ser progressivo (valor inicial < final)
! ou regressivo (valor inicial > final).

do elemento = 1, 5
! Este ciclo DO inicia com o valor 1 e termina com o valor 5, tendo como passo
! o valor unitário (por não ter sido especificado).
! É um ciclo do tipo progressivo, definido por variáveis inteiras.
! A variável "elemento" 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 "elemento" passará a ser a soma da variável
! "elemento" com "passo". Se o novo valor da variável "elemento" for menor ou
! igual a 5, 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(7,*) vetor(elemento)
    ! O comando READ é usado para ler cada "elemento" da variável "vetor" do
    ! arquivo 7 que corresponde ao arquivo "dados7a.txt", conforme definido no
    ! comando OPEN acima.
    ! Como há um asterisco, o conteúdo de cada variável é lido sem qualquer
    ! formato, e apenas uma variável por linha no arquivo "dados7a.txt".
    ! Note que só há um comando READ. Mas como ele está dentro de um ciclo, ele
    ! será suficiente para ler todos os elementos do vetor.

end do
! Encerra o ciclo DO

close(7)
! O comando CLOSE encerra o uso do arquivo 7

! -----

! Parte do programa onde são escritos os dados no arquivo de resultados

open(10, file='resultados7a.txt')
! Define o número "10" como uma referência ao arquivo "resultados7a.txt"

do elemento = 1, 5

    write(10,30) elemento, vetor(elemento)
    ! Escreve os valores das variáveis "elemento" e "vetor" no arquivo "10" com o
    ! formato especificado pelo número "30".
```

```

end do

! -----

! Parte do programa onde são realizadas operações empregando alguns comandos
! usados com vetores

maximo = maxval(vetor)
! O comando MAXVAL encontra o valor máximo de um vetor

local_maximo = maxloc(vetor)
! O comando MAXLOC determina o número do elemento do vetor que tem o
! valor máximo de um vetor

minimo = minval(vetor)
! O comando MINVAL encontra o valor mínimo de um vetor

local_minimo = minloc(vetor)
! O comando MINLOC determina o número do elemento do vetor que tem o
! valor mínimo de um vetor

soma = sum(vetor)
! O comando SUM realiza a soma dos valores de todos os elementos de um vetor

write(10,31) maximo, local_maximo, minimo, local_minimo, soma
! Escreve os valores de cinco variáveis no arquivo "10" com o
! formato especificado pelo número "31".

! -----

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

dos = systemqq('Notepad resultados7a.txt')
! Mostra o conteúdo do arquivo "resultados7a.txt" com o programa Notepad

fecha = setexitqq(qwin$exitnopersist)
! A instrução acima fecha automaticamente a janela do programa ao final de sua
! execução

! -----

! Parte do programa onde são definidos os formatos utilizados

30 format( 3x, 'vetor(', i1, ') =', i6 )

31 format (1/,'Valor máximo = ',i6, 1/, &
          'Elemento      = ',i6, 2/, &
          'Valor mínimo = ',i6, 1/, &
          'Elemento      = ',i6, 2/, &
          'Soma           = ',i6  )

! -----

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 “dados7a.txt”, inserindo nele os cinco elementos do vetor que são necessários para o programa7a.f90. Para fazer isso, execute o seguinte:
 - a) Start, Programs, Accessories, Notepad
 - b) Dentro do espaço de edição do Notepad, edite por exemplo:

```

0      elemento 1

```

```

5 elemento 2
-3 elemento 3
1 elemento 4
7 elemento 5
    
```

- c) File, Save
- d) Na janela “Save As”, em “File Name”, coloque “dados7a.txt”; em “Save in:”, selecione o diretório do seu projeto. Em seguida, clique sobre o botão “Save”. Finalmente, selecione File, Exit.
- 6) Execute o programa com Build, Execute. O resultado deverá ser:
- A abertura de uma janela do DOS.
 - A abertura de uma janela do WINDOWS que terá como título “Programa7a.f90 - UM VETOR COM 5 ELEMENTOS”, conforme foi especificado dentro do programa7a.f90.
 - A abertura do arquivo “dados7a.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 “dados7a.txt”. Depois, grave o novo valor com File, Save. Finalmente, feche este arquivo com File, Exit.
 - Em seguida, ocorrerá a abertura do arquivo “**resultados7a.txt**” com o programa Notepad. O conteúdo deverá ser:


```

vetor(1) = 0
vetor(2) = 5
vetor(3) = -3
vetor(4) = 1
vetor(5) = 7

Valor máximo = 7
Elemento = 5

Valor mínimo = -3
Elemento = 3

Soma = 10
            
```
 - Até entender, compare o conteúdo de cada linha do arquivo “resultados7a.txt” com os comandos do programa7a.f90 e seus respectivos formatos de edição.
 - Feche o arquivo “resultados7a.txt” com File, Exit.
 - Em seguida, tanto a janela do DOS quanto a do WINDOWS serão fechadas automaticamente.
- 7) Execute novamente o programa com Build, Execute. Utilize novos dados. Analise os resultados.

7.5 PROGRAMA7b.f90

- Clique dentro do espaço de edição do seu “programa7a.f90”. Depois execute: File, Close. Clique sobre o nome do “programa7a.f90”, no lado esquerdo da janela principal do Fortran. Em seguida, execute: Edit, Cut. Com isso, o “programa7a.f90” será eliminado do seu projeto, mas o arquivo continuará a existir dentro do diretório do seu projeto, podendo ser incluído nele quando desejado.
- Seguindo os passos da seção 7.3 deste tutorial, insira no projeto “programa07” o arquivo “programa7b.f90”.
- Dentro do espaço de edição de programas do Fortran, escreva o programa abaixo (para o programa funcionar, basta digitar apenas as linhas em vermelho, as demais são comentários que você deve ler e entender):

```

! -----
! Objetivo: definir um vetor com dimensão máxima pré-definida, com elementos do
!           tipo real, e realizar operações com ele
! -----
! Parte do programa onde são incluídas bibliotecas (módulos) do Fortran90
    
```

use msflib

! Inclui no programa a biblioteca do Fortran que permite usar os comandos
! "systemqq" e "setexitqq"

use msfwin

! Inclui no programa a biblioteca do Fortran que permite usar o
! comando "setwindowtext"

! -----

! 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"

logical janela

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

integer*4 fecha

! Declaração de variável do tipo "inteiro" para usar o comando "setexitqq"

real*8,dimension(5) :: vetor

! Declaração da variável chamada "vetor" como do tipo real dupla com
! cinco componentes, conforme especificado pelo comando "dimension(5)"

integer elemento

! Número da componente ou elemento do vetor

integer quantidade_elementos

! Número de componentes ou elementos do vetor

real*8 soma

! soma dos valores de cada componente do vetor

real*8 maximo

! valor máximo do vetor

real*8 minimo

! valor mínimo do vetor

integer local_maximo(1)

! número da componente ou elemento do vetor que tem o valor máximo do vetor

integer local_minimo(1)

! número da componente ou elemento do vetor que tem o valor mínimo do vetor

! -----

**janela = setwindowtext(gethwndqq(qwin\$framewindow), &
"Programa7b.f90 - UM VETOR COM ATÉ 5 ELEMENTOS"C)**

! Esta instrução define o nome da janela do programa.
! O nome deve ficar entre aspas, conforme o exemplo acima.

! -----

! Parte do programa onde são lidos os dados

dos = systemqq('Notepad dados7b.txt')

! Mostra o conteúdo do arquivo "dados7b.txt" com o programa Notepad

open(7,file='dados7b.txt')

! Define o número "7" como uma referência ao arquivo "dados7b.txt"

read(7,*) quantidade_elementos

! O comando READ é usado para ler a quantidade de elementos do vetor do
! arquivo 7 que corresponde ao arquivo "dados7b.txt", conforme definido no
! comando OPEN acima.

! Um CICLO na linguagem Fortran é definido por uma linha que inicia com o
! comando DO e outra linha que contém o comando END DO.

! Todas as linhas que estiverem entre os comandos DO e END DO serão repetidas
! de acordo com o que se define na linha do comando DO. Esta linha é definida
! por quatro variáveis:

! (1) um valor inicial do ciclo (no exemplo abaixo, usa-se o valor 1);
! (2) um valor final do ciclo (no exemplo abaixo, usa-se a variável
! "quantidade_elementos");
! (3) um passo para alterar a variável "contador" do ciclo (no exemplo abaixo,
! não foi especificado o valor do passo nem há uma variável para isso);
! quando não se define o valor do passo, ele vale 1;
! (4) um contador (no exemplo abaixo, usa-se a variável "elemento" como contador).
! Estas quatro variáveis podem ser do tipo inteiro (como neste exemplo) ou do
! tipo real.

! O ciclo pode ser progressivo (valor inicial < final)
! ou regressivo (valor inicial > final).

do elemento = 1, quantidade_elementos

! Este ciclo DO inicia com o valor 1 e termina com o valor da variável
! "quantidade_elementos", tendo como passo o valor unitário (por não ter sido
! especificado).

! É um ciclo do tipo progressivo, definido por variáveis inteiras.

! A variável "elemento" 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 "elemento" passará a ser a soma da variável
! "elemento" com "passo". Se o novo valor da variável "elemento" for menor ou
! igual à variável "quantidade_elementos", 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(7,*) vetor(elemento)

! O comando READ é usado para ler cada "elemento" da variável "vetor" do
! arquivo 7 que corresponde ao arquivo "dados7b.txt", conforme definido no
! comando OPEN acima.

! Como há um asterisco, o conteúdo de cada variável é lido sem qualquer
! formato, e apenas uma variável por linha no arquivo "dados7b.txt".

! Note que só há um comando READ. Mas como ele está dentro de um ciclo, ele
! será suficiente para ler todos os elementos do vetor.

end do

! Encerra o ciclo DO

close(7)

! O comando CLOSE encerra o uso do arquivo 7

! -----

! Parte do programa onde são escritos os dados no arquivo de resultados

open(10,file='resultados7b.txt')

! Define o número "10" como uma referência ao arquivo "resultados7b.txt"

do elemento = 1, quantidade_elementos**write(10,30) elemento, vetor(elemento)**

! Escreve os valores das variáveis "elemento" e "vetor" no arquivo "10" com o
! formato especificado pelo número "30".

end do

! -----

! Parte do programa onde são realizadas operações empregando alguns comandos
! usados com vetores

maximo = maxval(vetor)

! O comando MAXVAL encontra o valor máximo de um vetor

local_maximo = maxloc(vetor)

! O comando MAXLOC determina o número do elemento do vetor que tem o
! valor máximo de um vetor

minimo = minval(vetor)

! O comando MINVAL encontra o valor mínimo de um vetor

local_minimo = minloc(vetor)

! O comando MINLOC determina o número do elemento do vetor que tem o
! valor mínimo de um vetor

soma = sum(vetor)

! O comando SUM realiza a soma dos valores de todos os elementos de um vetor

write(10,31) maximo, local_maximo, minimo, local_minimo, soma

! Escreve os valores de cinco variáveis no arquivo "10" com o
! formato especificado pelo número "31".

! -----

close(10)

! O comando CLOSE encerra o uso do arquivo 10

dos = systemqq('Notepad resultados7b.txt')

! Mostra o conteúdo do arquivo "resultados7b.txt" com o programa Notepad

fecha = setexitqq(qwin\$exitnopersist)

! A instrução acima fecha automaticamente a janela do programa ao final de sua
! execução

! -----

! Parte do programa onde são definidos os formatos utilizados

30 format(3x, 'vetor(', i1, ') =', 1pe15.5)

**31 format (1/, 'Valor máximo = ', 1pe15.5, 1/, &
'Elemento = ', i6, 2/, &
'Valor mínimo = ', 1pe15.5, 1/, &
'Elemento = ', i6, 2/, &
'Soma = ', 1pe15.5)**

! -----

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 "dados7b.txt", inserindo nele a quantidade de elementos do vetor e os valores dos elementos que são necessários para o programa7b.f90. Para fazer isso, execute o seguinte:
 - a) Start, Programs, Accessories, Notepad
 - b) Dentro do espaço de edição do Notepad, edite por exemplo:


```
5      quantidade de elementos do vetor
0.0    elemento 1
5.0    elemento 2
-3.0   elemento 3
1.0    elemento 4
```


7.0 elemento 5

- c) File, Save
- d) Na janela “Save As”, em “File Name”, coloque “dados7b.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 que terá como título “Programa7b.f90 - UM VETOR COM ATÉ 5 ELEMENTOS”, conforme foi especificado dentro do programa7b.f90.
- c) A abertura do arquivo “dados7b.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 “dados7b.txt”. Depois, grave o novo valor com File, Save. Finalmente, feche este arquivo com File, Exit.
- d) Em seguida, ocorrerá a abertura do arquivo “resultados7b.txt” com o programa Notepad. O conteúdo deverá ser:
- ```

vetor(1) = 0.00000E+00
vetor(2) = 5.00000E+00
vetor(3) = -3.00000E+00
vetor(4) = 1.00000E+00
vetor(5) = 7.00000E+00

Valor máximo = 7.00000E+00
Elemento = 5

Valor mínimo = -3.00000E+00
Elemento = 3

Soma = 1.00000E+01

```
- e) Até entender, compare o conteúdo de cada linha do arquivo “resultados7b.txt” com os comandos do programa7b.f90 e seus respectivos formatos de edição.
- f) Feche o arquivo “resultados7b.txt” com File, Exit.
- g) Em seguida, tanto a janela do DOS quanto a do WINDOWS serão fechadas automaticamente.
- 9) Altere os dados do arquivo “dados7b.txt” para usar apenas 4 elementos, por exemplo:
- ```

4 quantidade de elementos do vetor
1.0 elemento 1
5.0 elemento 2
3.0 elemento 3
2.0 elemento 4

```
- 10) Execute novamente o programa com Build, Execute. O novo conteúdo do arquivo “resultados7b.txt” deverá ser:
- ```

vetor(1) = 1.00000E+00
vetor(2) = 5.00000E+00
vetor(3) = 3.00000E+00
vetor(4) = 2.00000E+00

Valor máximo = 5.00000E+00
Elemento = 2

Valor mínimo = 0.00000E+00
Elemento = 5

Soma = 1.10000E+01

```
- 11) Analise estes resultados. Perceba que todas as operações foram realizadas com os 5 elementos, que é a dimensão definida para o vetor com o comando **DIMENSION(5)**. Mas, como não foi lido do arquivo de dados o valor do 5º elemento, ele foi considerado igual a zero. Isso pode ser percebido pelo valor mínimo encontrado, que não está entre os 4 elementos lidos. Um problema semelhante ocorre quando se tenta usar mais elementos do que o definido pelo comando DIMENSION, conforme o exemplo a seguir.
- 12) Agora vamos ver o que ocorre quando se usa mais elementos do que foi definido. Altere os dados do arquivo “dados7b.txt” para:

```

6 quantidade de elementos do vetor
0.0 elemento 1
5.0 elemento 2
-3.0 elemento 3
1.0 elemento 4
7.0 elemento 5
20.0 elemento 6

```

- 13) Execute novamente o programa com Build, Execute. O novo conteúdo do arquivo “resultados7b.txt” deverá ser:

```

vetor(1) = 0.00000E+00
vetor(2) = 5.00000E+00
vetor(3) = -3.00000E+00
vetor(4) = 1.00000E+00
vetor(5) = 7.00000E+00
vetor(6) = 2.00000E+01

```

```

Valor máximo = 7.00000E+00
Elemento = 5

```

```

Valor mínimo = -3.00000E+00
Elemento = 3

```

```

Soma = 1.00000E+01

```

- 14) Analise estes resultados e perceba que não houve problema algum em ler e escrever 6 elementos do vetor, embora o último elemento já exceda o máximo que foi estipulado na definição do vetor com o comando **DIMENSION(5)**. Mas note que todas as operações com o vetor foram realizadas com apenas 5 elementos do vetor, que é o máximo definido no programa. Para evitar este problema, na próxima seção é apresentado o programa7c.f90.

- 15) **Conclusão: para que sejam realizadas corretamente operações com vetores, cada vetor deve ter sua dimensão corretamente estipulada pelo comando DIMENSION. Isto é, não se deve usar nem mais nem menos elementos no vetor do que o número definido com o comando DIMENSION para cada vetor.**

## 7.6 PROGRAMA7c.f90

- 1) Clique dentro do espaço de edição do seu “programa7b.f90”. Depois execute: File, Close. Clique sobre o nome do “programa7b.f90”, no lado esquerdo da janela principal do Fortran. Em seguida, execute: Edit, Cut. Com isso, o “programa7b.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 7.3 deste tutorial, insira no projeto “programa07” o arquivo “programa7c.f90”.
- 3) Dentro do espaço de edição de programas do Fortran, escreva o programa abaixo (para o programa funcionar, basta digitar apenas as linhas em vermelho, as demais são comentários que você deve ler e entender):

```

! -----
! Objetivo: definir um vetor com dimensão máxima alocável, com elementos do
! tipo real, e realizar operações com ele
! -----
! Parte do programa onde são incluídas bibliotecas (módulos) do Fortran90
use msflib
use msfwin
! -----

```

! Parte do programa onde são definidas as variáveis do programa

```
implicit none
```

```
logical dos
logical janela
integer*4 fecha
```

```
real*8,dimension(:),allocatable :: vetor
```

```
! Declaração da variável chamada "vetor" com as seguintes características:
! (1) comando REAL*8 a define como do tipo real dupla;
! (2) comando DIMENSION(:) a define com dimensão alocável ou aberta, conforme o
! comando ALLOCATABLE, isto é, sua dimensão será definida dentro do programa
! em função de alguma outra variável.
```

```
integer elemento
integer quantidade_elementos
real*8 soma
real*8 maximo
real*8 minimo
integer local_maximo(1)
integer local_minimo(1)
```

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

```
janela = setwindowtext(gethwndqq(qwin$framewindow), &
"Programa7c.f90 - UM VETOR COM NÚMERO INDEFINIDO DE ELEMENTOS"C)
```

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

! Parte do programa onde são lidos os dados

```
dos = systemqq('Notepad dados7c.txt')
```

```
open(7,file='dados7c.txt')
```

```
read(7,*) quantidade_elementos
```

```
allocate (vetor(quantidade_elementos))
```

```
! Com este comando ALLOCATE define-se quantos elementos a variável "vetor" terá.
! Note que esta definição ocorre em função da variável "quantidade_elementos",
! que é um dado do programa. Portanto, a alocação de memória para a variável
! "vetor" não é pré-definida, ela ocorre em função de dados do programa.
```

```
do elemento = 1, quantidade_elementos
```

```
 read(7,*) vetor(elemento)
```

```
end do
```

```
close(7)
```

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

! Parte do programa onde são escritos os dados no arquivo de resultados

```
open(10,file='resultados7c.txt')
```

```
do elemento = 1, quantidade_elementos
```

```
 write(10,30) elemento, vetor(elemento)
```

```
end do
```

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

! Parte do programa onde são realizadas operações empregando alguns comandos  
! usados com vetores

```

maximo = maxval(vetor)

local_maximo = maxloc(vetor)

minimo = minval(vetor)

local_minimo = minloc(vetor)

soma = sum(vetor)

write(10,31) maximo, local_maximo, minimo, local_minimo, soma

! -----

close(10)

dos = systemqq('Notepad resultados7c.txt')

fecha = setexitqq(qwin$exitnopersist)

! -----

! Parte do programa onde são definidos os formatos utilizados

30 format(3x, 'vetor(', i1, ') =', lpe15.5)

31 format (1/, 'Valor máximo = ', lpe15.5, 1/, &
 'Elemento = ', i6, 2/, &
 'Valor mínimo = ', lpe15.5, 1/, &
 'Elemento = ', i6, 2/, &
 'Soma = ', lpe15.5)

! -----

end

```

- 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 “dados7c.txt”, inserindo nele a quantidade de elementos do vetor e os valores dos elementos que são necessários para o programa7c.f90. Para fazer isso, execute o seguinte:
  - a) Start, Programs, Accessories, Notepad
  - b) Dentro do espaço de edição do Notepad, edite por exemplo:
 

```

4 quantidade de elementos do vetor
1.0 elemento 1
5.0 elemento 2
3.0 elemento 3
2.0 elemento 4

```
  - c) File, Save
  - d) Na janela “Save As”, em “File Name”, coloque “dados7c.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 que terá como título “Programa7c.f90 - UM VETOR COM NÚMERO INDEFINIDO DE ELEMENTOS”, conforme foi especificado dentro do programa7c.f90.
  - c) A abertura do arquivo “dados7c.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 “dados7c.txt”. Depois, grave o novo valor com File, Save. Finalmente, feche este arquivo com File, Exit.

d) Em seguida, ocorrerá a abertura do arquivo “resultados7c.txt” com o programa Notepad. O conteúdo deverá ser:

```
vetor (1) = 1.00000E+00
vetor (2) = 5.00000E+00
vetor (3) = 3.00000E+00
vetor (4) = 2.00000E+00

Valor máximo = 5.00000E+00
Elemento = 2

Valor mínimo = 1.00000E+00
Elemento = 1

Soma = 1.10000E+01
```

e) Até entender, compare o conteúdo de cada linha do arquivo “resultados7c.txt” com os comandos do programa7c.f90 e seus respectivos formatos de edição.

f) Feche o arquivo “resultados7c.txt” com File, Exit.

g) Em seguida, tanto a janela do DOS quanto a do WINDOWS serão fechadas automaticamente.

9) Altere os dados do arquivo “dados7c.txt” para usar o número de elementos que você quiser. Execute novamente o programa com Build, Execute.

**10) RECOMENDAÇÃO: para evitar qualquer problema no uso de vetores, deve-se usar o comando DIMENSION(:),ALLOCATABLE que permite definir dentro do próprio programa a dimensão de cada vetor de acordo com variáveis a serem lidas do arquivo de dados. Assim, cada vetor é dimensionado para o conjunto de dados usado em cada execução do programa.**

## **7.7 FECHANDO UM PROJETO**

- 1) Dentro da “janela principal” do Fortran, feche o seu projeto que está aberto. Para fazer isso, basta executar o seguinte: File, Close Workspace
- 2) Na janela “Microsoft Developer Studio” clique sobre o botão “Yes”.

## **7.8 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 sair do Windows, por favor, apenas desligue o monitor do computador.

## LISTA DE EXERCÍCIOS DA 7ª AULA

(Para ser entregue na próxima aula)

### Exercício 7.1

- 1) Com base no programa7c.f90, implemente um programa em Fortran para ler e escrever dois vetores com dimensões diferentes e alocação. Utilize o seguinte algoritmo:
  - a) Ler do arquivo “dados7.txt”: o número de elementos do primeiro vetor; os valores dos elementos do primeiro vetor; o número de elementos do segundo vetor; os valores dos elementos do segundo vetor.
  - b) Escrever no arquivo “resultados7.txt” todos os valores lidos do arquivo “dados7.txt”.
- 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, do arquivo “dados7.txt” e do arquivo “resultados7.txt”

### Exercício 7.2

- 1) Acesse o site da disciplina no endereço:  
[ftp://ftp.demec.ufpr.br/Disciplinas/Tm102/Marchi/semestre\\_2003\\_2/](ftp://ftp.demec.ufpr.br/Disciplinas/Tm102/Marchi/semestre_2003_2/)
- 2) Estude as páginas 32 a 34 do capítulo 7 da “apostila\_logica\_programacao.pdf”.