

# Control de Fortran

---

Crea o arquivo `datos1.dat` coas dúas columnas de números enteiros:

```
8 8
4 3
5 5
3 3
4 4
7 5
7 7
8 5
9 1
6 6
1 4
```

Escribe un programa en Fortran que defina un subprograma chamado `contarCoincidencias()`, co tipo e argumentos axeitado, que lea os datos do arquivo `datos1.dat` liña a liña. Ademáis, debe sumar os valores de todas as liñas nas que coinciden os seus números ata que a suma chegue a 50. O subprograma ten que devolver o número de liñas sumadas. Sexa  $n$  o número de liñas sumadas.

O programa principal debe chamar o subprograma `contarCoincidencias()`, e crear un vector  $\mathbf{v}$  de dimensión  $n$ , onde cada elemento  $v_i$ , con  $i = 1, \dots, n$ , é:

$$v_i = e^{2i} \operatorname{sen} \left( \frac{2\pi i}{3} \right) \quad (1)$$

Posteriormente, o programa principal debe mostrar por pantalla o vector  $\mathbf{v}$  nunha única liña con 2 cifras decimais. NOTA: debes obter  $n = 5$  e o vector  $\mathbf{v} = 6.40 \ -47.28 \ -0.00 \ 2581.59 \ -19075.38$ .

```
program controll
integer :: contarCoincidencias
real, dimension(:), allocatable :: v
real, parameter :: pi = 3.14159
n=contarCoincidencias()
print *, "Numero de linhas: ", n
allocate(v(n))
do i=1,n
    v(i)=exp(real(2*i))*sin(2*pi*i/3)
end do
print '(a$)', "v="
do i=1,n
    print '(f10.2_)$', v(i)
end do
deallocate(v)
stop
end program controll

integer function contarCoincidencias()
integer :: suma
nlinhas=0
open(1, file="datos1.dat", status="old", err=3)
do
    read(1,*, end=4) n, m
    if (n == m) then
        suma = suma + n + m
        nlinhas = nlinhas + 1
        if (suma >= 50) exit
    end if
end do
```

```
    end if
end do
4 close(1)
contarCoincidencias = nlinhas
return
3 print*, "Erro_abrindo_datos1.dat"
end function contarCoincidencias
```

## Control de Fortran

---

Crea o arquivo `datos2.dat` que almacena unha matriz de números enteiros de orde  $n$ , onde a primeira liña contén a orde da matriz:

```
5
1 3 2 5 4
6 4 3 2 4
5 4 3 6 1
4 8 9 6 4
5 3 4 1 2
```

Escrebe un programa que lea o arquivo `datos2.dat` e almacene os datos na matriz **a**. Define un subprograma chamado `calculaVector()`, cos argumentos e tipo axeitado, que calcule un vector **v** de lonxitude  $n$  (a orde da matriz) cos elementos  $v_i$  dados por:

$$v_i = \sum_{j=1}^i \sum_{k=1}^i a_{jk} \quad 1 \leq i \leq n \quad (2)$$

O programa principal ten que chamar ó subprograma `calculaVector()` para calcular o vector **v** e visualizalo na pantalla. Posteriormente, debe visualizar por pantalla os elementos de **v** mentres que a suma sexa menor que 50. NOTA: debes obter o vector **v**=1 14 31 71 99 e posteriormente, ten que mostrar os valores 1 14 31.

```
program control2
integer, dimension(:, :), allocatable :: a
integer, dimension(:), allocatable :: v
integer :: suma
open(2, file="datos2.dat", status="old", err=3)
read(2, *) n
allocate(a(n,n), v(n))
do i=1,n
    read(2, *) (a(i, j), j=1,n)
end do
call calculaVector(a, n, v)
print *, "Vector_v=", (v(i), i=1,n)
print '(a,$)', " Valores_de_v:"
suma=0
i=1
do
    suma= suma + v(i)
    if (suma < 50) then
        print '(i4,$)', v(i)
        i=i+1
    else
        exit
    end if
end do
deallocate(a, v)
stop
3 print *, "Erro_labrindo_arquivo_datos2.dat"
end program control2
```

```
subroutine calculaVector(a, n, x)
integer, dimension(n,n), intent(in) :: a
integer, intent(in) :: n
```

```
integer, dimension(n), intent(out)::x
do i=1,n
  x(i)=0
  do j=1,i
    do k=1,i
      x(i) = x(i) + a(j,k)
    end do
  end do
end do
return
end subroutine calculaVector
```

## Control de Fortran

---

Escribe un programa en Fortran que declare un vector estático de números enteros  $v$  con 100 elementos. O programa ten que ler números por teclado ata que a súa suma sexa maior que 30 e almacenalos no vector  $v$ . Escribe un subprograma chamado `calculaVector()`, cos argumentos e tipo axeitado, que devolva un vector  $y$  cos valores de  $v$  no intervalo  $[5, 10]$ . O programa principal ten que chamar ó subprograma `calculaVector()` e, posteriormente, gardar no arquivo `exame3.dat` os vectores  $v$  e  $y$  (un vector en cada liña).

NOTA: se introduces 3,2,7,1,6,5,10,9,7 o contido do arquivo `exame3.dat` sería:

```
v= 3 2 7 1 6 5 10
y= 7 6 5 10
```

```
program control3
integer, dimension(100)::v, y
open(1, file='exame3.dat', status='new', err=1)
n=0
suma=0
do
  print '(a,$)', "No:_"
  read*, v(n)
  suma = suma + v(n)
  n=n+1
  if (suma > 30) exit
end do
write (1,*) "vector_v:_" , (v(i), i=1,n)
call calculaVector(v, n, y, ny)
write (1,*) "vector_y:_" , (y(i), i=1,ny)
close(1)
stop
1 print *, 'erro_en_open_abrindo_exame3.dat'
stop
end program control3
```

```
subroutine calculaVector(v,n,y, ny)
integer, dimension(100), intent(in)::v
integer, intent(in)::n
integer, dimension(100), intent(out)::y
integer, intent(out)::ny
ny=0
do i=1,n
  if(v(i) >=5 .and. v(i) <= 10) then
    ny=ny+1
    y(ny)=v(i)
  end if
end do
return
end subroutine calculaVector
```

## Control de Fortran

---

Crea co editor de texto un arquivo chamado `datos5.dat` co seguinte contido:

```
8
3 1 0 8 2 7 9 5
```

onde a primeira liña é o número  $n$  de elementos na segunda liña. Escribe un programa en Fortran que lea o arquivo `datos5.dat` e o almacene no vector  $\mathbf{v}$  de dimensión  $n$ . Define un subprograma chamado `convirte()`, cos argumentos e tipo axeitado, que calcule un vector  $\mathbf{w}$ , da mesma lonxitude  $n$  que  $\mathbf{v}$ , de

modo que o elemento  $w_i$ , con  $1 \leq i \leq n$ , sexa: se  $i$  é múltiplo de 3, entón  $w_i = \sum_{j=1}^i j^2 v_j$ ; en caso contrario,  $w_i = i^2$ .

O programa principal debe chamar ó subprograma `convirte()`, e logo visualizar na terminal o vector  $\mathbf{w}$ , informando se o vector  $\mathbf{w}$  ten algún elemento repetido ou non (non mostres os elementos repetidos).

**NOTA:** debes obter  $\mathbf{w}=1,4,7,16,25,437,49,64$ , que non ten elementos repetidos.

```
program control
integer , dimension (:), allocatable :: v, w
logical :: repetido=.false.
open(7, file="datos5.dat", status="old", err=2)
read(7,*) n
allocate(v(n), w(n))
read(7,*)(v(i), i=1,n)
close(7)
print *, "vector v: ", (v(i), i=1,n)
call convertir(v, n, w)
print *, "vector w: ", (w(i), i=1,n)
filas: do i=1,n
    k=w(i)
    do j=i+1,n
        if(w(j)==k) then
            repetido=.false.
            exit filas
        end if
    end do
end do filas
if(repetido) then
    print *, 'w ten elementos repetidos'
else
    print *, 'w non ten elementos repetidos'
end if
deallocate(v,w)
stop
2 print *, "Erro abrindo o arquivo datos5.dat"
end program control
```

```
subroutine convertir(x,n,y)
integer , dimension(n), intent(in)::x
integer , intent(in)::n
integer , dimension(n), intent(out)::y
do i=1,n
    if(0==mod(i,3)) then
        s=0
        do j=1,i
            s=s+j*j*x(j)
        end do
        y(i)=s
    end if
end do
```

```
        end do
        y(i)=s
    else
        y(i)=i*i
    end if
end do
return
end subroutine convertir
```