

Informática. Exame de Prácticas.

Febreiro, 2007

Nome:

Escribir un programa en Fortran que faga o seguinte:

1. (1 PUNTO) Declarar tres vectores **x**, **y** e **z** dinámicos con valores reais. Pedir ao usuario e ler por teclado un número enteiro *n* e reservar memoria dinamicamente para os tres vectores. Esta memoria debe ser liberada ao remata-lo programa.
2. (3 PUNTOS) Chamar a un subprograma `le_vectores(...)` (tes que decidi-lo seu tipo e argumentos), que lea as compoñentes dos dous primeiros vectores dende o arquivo `vectores.dat`, almacenadas segundo o formato:

```
x1 y1
x2 y2
x3 y3
...
```

3. (4 PUNTOS) Chamar a un subprograma `calcula_convolucion(...)` (tes que decidi-lo seu tipo e argumentos), que calcule as compoñentes do vector **z** segundo a expresión:

$$z_i = \sum_{j=1}^{n-i} x_j y_{j+i} \quad i = 1, \dots, n \quad (1)$$

4. (2 PUNTOS) Chamar ao suprograma `almacena(...)` (tes que decidi-lo seu tipo e argumentos) que almacene no arquivo `convolucion.dat` os tres vectores por columnas (un en cada columna).

SOLUCIÓN

```
program convolucion
real, dimension(:), allocatable :: x, y, z

print *, "introduza n: "
read *, n
allocate(x(n), y(n), z(n))

call le_vectores(x, y, n)
call calcula_convolucion(x, y, z, n)
call almacena(x, y, z, n)

deallocate(x, y, z)
stop
end program convolucion
```

cc

```

subroutine le_vectores(x, y, n)
real, dimension(n), intent(out) :: x, y
integer, intent(in) :: n

open(1, file = "vectores.dat", status = "old", err = 1)
do i = 1, n
  read (1, *) x(i), y(i)
end do
close(1)

return
1 print *, "le_vectores: erro en open"
stop
end subroutine le_vectores

```

```

cccccccccccccccccccccccccccccccccccccccc
subroutine calcula_convolucion(x, y, z, n)
real, dimension(n), intent(in) :: x, y
real, dimension(n), intent(out) :: z
integer, intent(in) :: n

do i = 1, n
  z(i) = 0
  do j = 1, n - i
    z(i) = z(i) + x(j)*y(j + i)
  end do
end do

return
end subroutine calcula_convolucion

```

```

cccccccccccccccccccccccccccccccccccccccc
subroutine almacena(x, y, z, n)
real, dimension(n), intent(in) :: x, y, z
integer, intent(in) :: n

open(1, file = "convolucion.dat", status = "new", err = 1)
do i = 1, n
  write (1, *) x(i), y(i), z(i)
end do
close(1)

return
1 print *, "almacena: erro en open"
stop
end subroutine almacena

```