

Control Fortran

Escribe un programa en Fortran que lea por teclado un número inteiro n , reserve dinámicamente memoria para un vector inteiro \mathbf{v} de dimensión n e lea por teclado as súas compoñentes. Logo, debe chamar a un subprograma `calcula_matriz(...)` (debes decidi-lo seu tipo e argumentos) que calcule unha matriz cadrada de orde n tal que a súa **diagonal secundaria** sexa igual ao vector \mathbf{v} . Os restantes elementos da matriz deben te-lo valor -1. Finalmente, dende o programa principal debes chamar a outro subprograma `almacena_matriz(...)` (debes decidi-lo seu tipo e argumentos) que cree o arquivo `matriz.dat` e almacene nel a matriz (cada fila debe escribirse nunha liña distinta do arquivo). Proba con $n = 4$, $\mathbf{v} = [1, 2, 3, 4]$; debes obte-la matriz:

$$\begin{bmatrix} -1 & -1 & -1 & 1 \\ -1 & -1 & 2 & -1 \\ -1 & 3 & -1 & -1 \\ 4 & -1 & -1 & -1 \end{bmatrix}$$

SOLUCIÓN:

```
program diagonal_secundaria

real, dimension(:), allocatable::v
real, dimension(:,:), allocatable::a
print*, "Introduce un numero: "
read*, n
allocate(v(n), a(n,n))
print*, "Introduce vector: "
read*, (v(i), i=1,n)
call calcula_matriz(v, a, n)
call almacena_matriz(a, n)
deallocate(v, a)
stop
end program diagonal_secundaria

! Definicion de calcula_matriz
subroutine calcula_matriz(x, b, m)
real, dimension(m), intent(in)::x
real, dimension(m,m), intent(out)::b
integer, intent(in)::m
do i=1, m
  do j=1,m
    if (i+j == m+1) then
      b(i,j)=x(i)
    else
      b(i,j)=-1
    end if
  end do
end do
return
end subroutine calcula_matriz
```

```
!Definicion de almacena_matriz
subroutine almacena_matriz(c, m)
real, dimension(m,m), intent(in)::c
integer, intent(in)::m
open(1, file="matriz.dat", status="new", err=3)
do i=1, m
  write(1,*) (c(i,j), j=1,m)
end do
close(1)
return
3 print*, "Erro abrindo matriz.dat"
end subroutine almacena_matriz
```