

## Control de Fortran

---

Crea un archivo chamado `datos_exame.dat` co seguinte contido (cada número nunha fila):

```
8 2 1 4 9
```

Escribe un programa chamado `exame.f90` que declare un vector **x** e unha matriz **a** dinámicos con valores enteiros e abra o anterior arquivo `datos_exame.dat`, conte o número  $n$  de liñas e reserve memoria para **x** e **a**, ambos de orde  $n$ . Logo, o programa debe ler os valores no arquivo e almacenalos no vector **x**, mostrando por pantalla **x** nunha única liña. Despois, debe chamar ao subprograma `matriz(...)`, do tipo e cos argumentos axeitados, que calcule **a** de modo que  $a_{ij}$ , con  $i, j = 1, \dots, n$ , sexa o número de valores enteiros, comezando en 1 e aumentando de 1 en 1, que hai que sumar para superar  $i^2j$ . Finalmente, o programa principal debe mostrar a matriz **a** na terminal, unha fila en cada liña da terminal, cada elemento co ancho mínimo necesario.

```
program exame
integer , allocatable :: x(:) , a(:, :)
n=0
open(1, file='datos_exame.dat', status='old', err=1)
do
    read (1, *, end=2); n=n+1
end do
2 allocate(x(n), a(n, n)); rewind(1)
do i=1, n
    read (1, *) x(i)
end do
close(1)
print *, 'x=', x
call matriz(a, n)
print *, 'a='
do i=1, n
    do j=1, n
        print '(i0," ", $)', a(i, j)
    end do
    print *, ''
end do
deallocate(x, a)
stop
1 stop 'archivo datos_exame.dat non atopado'
end program exame
```

```
!-----
subroutine matriz(a, n)
integer , intent(out) :: a(n, n)
integer , intent(in) :: n
integer :: p
do i=1, n
    do j=1, n
        p=i**2*j; k=1; l=0
        do
```

```
        l=l+k;k=k+1
        if(l>p) exit
    end do
    a(i,j)=1
end do
end do
end subroutine matriz
```