

Control de Fortran

Escribe un programa en fortran chamado `exame1.f90` que lea por teclado un número enteiro positivo n (usa $n = 5$). Crea un vector \mathbf{p} de dimensión n , onde cada elemento do vector p_i , $i = 1, \dots, n$ calcúlase co subprograma `calculaValor()` (que debes definir cos argumentos axeitados) e que se define como:

$$p_i = \sum_{n=0}^i n \cos\left(\frac{n\pi}{10}\right)$$

No programa principal, calcula unha matriz \mathbf{a} cadrada de orde n , onde cada elemento a_{ij} sexa o número de elementos do vector \mathbf{p} (comezando polo índice 1 e volvendo a comezar polo índice 1 cando se chegue ó final do vector) para que se cumpra a ecuación:

$$\sum_k p_k < ij^2$$

Garda no arquivo `resultado.txt` a matriz \mathbf{a} . NOTA: para $n = 5$ o vector \mathbf{p} sería:

```
p= 0.951056600      2.56909132      4.33244896      5.56852102      5.56852722
```

SOLUCIÓN:

```
program exame1
real ,dimension (:,:), allocatable :: a
real ,dimension (:), allocatable :: p
print '("n= ", $)'
read*, n
allocate(p(n), a(n,n))
do i=1,n
    p(i)=calculaValor(i)
end do
print*, "p= ", p
do i=1,n
    do j=1,n
        numbral=i*j*j
        nv=0
        suma=0
        k=1
        do
            suma= suma + p(k)
            k=k+1
            if(k>n) k=1
            nv= nv +1
            if (suma > numbral) exit
        end do
        a(i,j)=nv
    end do
end do
open(1, file="resultados.txt", status="new", err=2)
do i=1,n
    write(1,*)(a(i,j), j=1,n)
end do
close(1)
stop
2 print*, "Erro escribindo en resultados.txt"
end program exame1

!*****
function calculaValor(m)
integer, intent(in) :: m
```

```
real ,parameter :: pi=3.14159
f(n)=n*cos(pi*n/10)
suma= 0
do i=0, m
    suma = suma + f(i)
end do
calculaValor=suma
return
end function calculaValor
```

Control de Fortran

Escribe un programa en fortran chamado `exame2.f90` que lea por teclado un número enteiro positivo n maior que 3. O programa non debe rematar mentres o usuario introduza un número incorrecto. Crea un vector \mathbf{p} de dimensión n , onde cada elemento p_i , $i = 1, \dots, n$ ven dado pola expresión:

$$p_i = \sum_{j=0}^{k-1} \frac{5^j}{k^2} \cos\left(\frac{i\pi j}{20k}\right) \quad k = 100$$

Define un subprograma chamado `calculos()`, decide os argumentos axeitados, que que calcule un vector \mathbf{y} dado por: $y_1 = 1$ e $y_i = p_i - p_{i-1}$, $i = 2, \dots, n$.

Desde o programa principal chama ó subprograma `calculos()` e garda no arquivo `resultados.txt` os vectores \mathbf{p} (co formato por defecto) e \mathbf{y} (como reais con expoñente).

NOTA: con $n = 5$ o contido do arquivo `resultados.txt` sería:

```
p=      2.45990586      2.41486955      2.34062266      2.23836994      2.10976601
y=      0.10E+01     -0.45E-01     -0.74E-01     -0.10E+00     -0.13E+00
```

SOLUCIÓN:

```
program exame1
real , dimension (:), allocatable :: p, y
real , parameter :: pi=3.14159
do
  print '("n= ", $)'
  read*, n
  if (n > 3) exit
end do
allocate(p(n), y(n))
k=100
do i=1,n
  t=0
  do j=0, k-1
    t= t + 5*j*cos(i*pi*j/20/k)/k**2
  end do
  p(i)= t
end do
call calculos(p, y, n)
open(1, file="resultados.txt", status="new", err=4)
write(1, '("p= ", $)')
write(1, *) (p(i), i=1,n)
write(1, '("y= ", $)')
do i=1,n
  write(1, '(e12.2, t20, $)') y(i)
end do
close(1)
deallocate(y,p)
stop
4 print*, "Erro abrindo resultados.txt"
end program exame1

!*****
subroutine calculos(x, z, n)
real , dimension(n), intent(in) :: x
real , dimension(n), intent(out):: z
integer , intent(in) :: n
z(1)=1
do i=2,n
  z(i)=x(i)-x(i-1)
end do
```

```
return  
end subroutine calculos
```

Control de Fortran

Co editor de texto crea o arquivo `datos3.txt` con un número enteiro en cada liña. Por exemplo cos números 2, 3, 1, 1, 2, 5, 3, 3, 3, 5, 5, 7, 7, 8, 1.

Escribe un programa en fortran chamado `exame3.f90` que defina o subprograma `analizaArquivo()`, debes decidir os argumentos axeitados, que lea o arquivo `datos3.txt` e calcule: 1) a suma dos números sempre que coincidan (sexan o mesmo) en dúas liñas consecutivas (debe sumar números mentres que a suma sexa menor que 10 ou se remate o arquivo); 2) o número de elementos sumados.

O programa principal ten que chamar ó subprograma `analizaArquivo()` e visualizar na pantalla o valor da suma (s) e o número de elementos sumados(n). Logo, ten que crear unha matriz \mathbf{a} de orde n definida por:

$$a_{ij} = \frac{(ji)^2}{\sqrt{(j-i)^2 + 1}} \quad i, j = 1, \dots, n$$

Visualiza na pantalla a matriz \mathbf{a} mostrando unha fila da matriz en cada liña.

NOTA: utilizando os datos do exemplo obteríase un $s = 12$ e un $n = 4$, ya que se sumarían os números 1, 3, 3, 5.

SOLUCIÓN:

```
program exame3
integer :: s
real, dimension(:, :), allocatable :: a
real, parameter :: pi=3.14159
call analizaArquivo(s, n)
print *, "Suma= ", s, " Numero: ", n
allocate(a(n,n))
do i=1,n
  do j=1,n
    a(i, j)= float((i*j)**2)/sqrt((float(j-i)**2+1))
  end do
end do
print *, "Matriz a:"
do i=1,n
  print *, (a(i, j), j=1,n)
end do
deallocate(a)
stop
end program exame3

!*****
subroutine analizaArquivo(suma, numero)
integer, intent(out) :: suma, numero
open(1, file="datos3.txt", status="old", err=3)
suma=0; numero=0
read(1, *, end=4) n
do
  if (suma >= 10) then
    close(1)
    exit
  end if
  read(1, *, end=4) m
  if( m == n) then
    suma = suma + n
    numero= numero + 1
  end if
  n=m
end do
4 close(1)
```

```
return  
3 print*, "Erro abrindo datos3.txt"  
end subroutine analizaArquivo
```

Control de Fortran

Escribe un programa en fortran llamado `exame4.f90` que pida ó usuario un número real a e cree un vector de números enteros \mathbf{x} con $n = 30$ elementos, x_i dado por:

$$x_i = \left\lfloor 10 \cos^2 \left(\frac{a\pi i}{4} \right) \right\rfloor \quad i = 1, \dots, n$$

Define un subprograma chamado `contarOcurrencias()`, cos argumentos axeitados, que dado un número enteiro m e un vector \mathbf{z} , calcule o número de veces que se repite m no vector z . No programa principal, calcula o valor máximo de \mathbf{x} (mx) e crea un vector \mathbf{v} de dimensión $mx + 1$ onde cada elemento v_i , $i = 0, \dots, mx$ calcúlase chamando repetidamente ó subprograma `contarOcurrencias()` e será o número de veces que se repite o número i no vector \mathbf{x} . Calcula tamén o número de elementos do vector \mathbf{x} que hai que sumar para que a suma supere o valor 1000, tendo en conta que cando se chegue ó final do vector vólvese a comezar no principio. Garda este valor no arquivo `resultados4.txt` e os vectores \mathbf{x} e \mathbf{v} .

NOTA: se utilizas $a = 2.5$ obtés o seguintes resultados:

```
Numero elementos sumados: 220
 1 5 8 0 8 4 1 10 1 5 8 0 8 4 1 10 1 5 8 0 8 4 1 10 1 5 8 0 8 4
 4 7 0 0 4 4 0 0 8 0 3
```

SOLUCIÓN:

```
program exame4
integer , dimension (30):: x
integer , dimension (:), allocatable :: v
integer :: contarOcurrencias
real , parameter :: pi=3.14159
print '("a= ", $)'
read *, a
n=30
mx=0
do i=1, n
    x(i)=int(10*cos(a*pi*i/4)**2)
    if(x(i) > mx) mx=x(i)
end do
allocate(v(0:mx))
do i=0, mx
    v(i)=contarOcurrencias(x, n, i)
end do
suma=0
i=1
nc= 0
do
    suma = suma + x(i)
    nc= nc + 1
    if (suma > 1000) exit
    i= i+ 1
    if( i > n) i=1
end do
open(2, file="resultados4.txt", status="new", err=8)
write(2, '("Numero elementos sumados: ", i0)') nc
write(2, *) (x(i), i=1,n)
write(2, *) (v(i), i=0,mx)
close(2)
deallocate(v)
stop
8 print *, "Erro abrindo arquivo resultados4.txt"
end program exame4
```

```
!*****
```

```
integer function contarOcuurrencias(z,n, m)
integer , dimension(n), intent(in) :: z
integer , intent(in) :: n, m
k=0
do i=1,n
    if (z(i) == m) k= k+1
end do
contarOcuurrencias=k
return
end function contarOcuurrencias
```


Control de Fortran

Utilizando o editor de texto crea o arquivo `datos5.txt` que contén a dimensión e os valores de dous vectores: x de dimensión m e y de dimensión n (primeira liña m , segunda liña x , terceira liña n e cuarta liña y) con $m < n$. Por exemplo, `datos5.txt` podería conter os datos:

```
3
3 6 4
10
1 3 4 2 5 8 3 6 4 7
```

Escrebe un programa en fortran chamado `exame5.f90` que visualice na pantalla os vectores x e y e amose unha mensaxe informando se o vector x está contido ou non no vector y utilizando o subprograma `estaContido()`. Este subprograma tes que definilo ti cos argumentos axeitados.

Calcula a suma dos elementos dos vectores x e y que serán respectivamente sx e sy . Calcula tamén o número de veces que hai que dividir por 2 o valor sy para que sy sexa menor que sx .

NOTA: co exemplo anterior terías que obter que x está contido en y , e o número de veces sería 2.

SOLUCIÓN:

```
program exame5
integer , dimension (:), allocatable :: x, y
logical :: estaContido
open(8, file="datos5.txt", status="old", err=1)
read(8,*) m
allocate(x(m))
read(8,*) (x(i), i=1,m)
read(8,*) n
allocate(y(n))
read(8,*) (y(i), i=1,n)
close(8)
print*, "x= ", (x(i), i=1,m)
print*, "y= ", (y(i), i=1,n)

if (estaContido(x,m,y,n)) then
    print*, "Vector x contido no vector y"
else
    print*, "Vector x non contido en vector y"
end if
sx=sum(x)
sy=sum(y)
nv=0
do
    if (sy < sx) exit
    sy = sy/2
    nv=nv +1
end do
print*, "Numero de veces: ", nv
deallocate(x,y)
stop
! print*, "Erro abrindo arquivo datos5.txt"
end program exame5

!*****
logical function estaContido(x,m, y, n)
integer , dimension(m), intent(in) :: x
integer , dimension(n), intent(in) :: y
integer , intent(in) :: n, m
estaContido=.false.
do i=1, n-m
```

```
estaContido=.true.  
do j=1, m  
  if (x(j) /= y(i+j)) then  
    estaContido= .false.  
    exit  
  end if  
end do  
if (estaContido) return  
end do  
return  
end function estaContido
```