

Exame de Programación Científica, 1º Matemáticas

xaneiro, 2026, grupo CLE1

NOTA: debes obter alomenos 1 punto en cada exercicio.

(5 PUNTOS) Escribe un programa `exame.f90` en Fortran que defina $n=10$. Define un vector $\mathbf{x}$ con n valores aleatorios enteiros entre 1 e n , usando a sentenza $\mathbf{x}(i)=\text{int}(n*\text{rand}()+1)$ para $i = 1..n$. Define outro vector $\mathbf{y}$ de lonxitude n con elementos $y_i = y_{i-1} - x_{i-1}$ para i par e $y_i = 2x_i$ para i impar. Mostra $\mathbf{x}$ e $\mathbf{y}$ por pantalla, cada un nunha liña. Chama ao subprograma `subprog(·)` que defina unha matriz $\mathbf{a}$ cadrada de orde n . O elemento a_{ij} para $i, j = 1..n$ debe ser a suma s dos valores $x_k/(1+y_k^2)$ comezando en $k = 1 + (i+j) \% n$, onde $\%$ é o resto da división enteira, volvendo ao principio cando

$k > n$ e rematando cando s supere o umbral $u = \frac{\sum_{l=1}^i x_l}{1 + \sum_{l=1}^j y_l}$. No programa principal, escribe $\mathbf{x}$ no

arquivo `fortran.txt`, nunha liña con formato enteiro de ancho mínimo, e tamén $\mathbf{a}$, cada fila nunha liña, con formato real de ancho 5 e 1 decimal.

```
program exame
integer,parameter :: n=10
integer :: x(n),y(n),z
real :: a(n,n)
do i=1,n
  x(i)=int(n*rand()+1)
  if(mod(i,2)==0) then
    j=i-1;y(i)=y(j)-x(j)
  else
    y(i)=2*x(i)
  end if
end do
print *, 'x=', x
print *, 'y=', y
call subprog(x,y,n,a)
open(1,file='fortran.txt')
write(1, '( "x=", $ ) ')
write(1, '( i0, " ", $ ) ') x; write(1,*)
write(1,*) 'a='
do i=1,n
  write(1, '( f5.1, " ", $ ) ') a(i,:); write(1,*)
end do
end program exame
!-----
subroutine subprog(x,y,n,a)
integer,intent(in) :: x(n),y(n),n
real,intent(out) :: a(n,n)
sx=0
do i=1,n
  sx=sx+x(i);sy=0
  do j=1,n
    k=1+mod(i+j,n);sy=0
    sy=sy+y(j);u=sx/(1+sy)
    do
      s=s+x(k)/(1.+y(k)**2)
      if(s>u) exit
      k=k+1
      if(k>n) k=1
    end do
  end do
end do
```

```

        a(i,j)=s
    end do
end do
end subroutine subprog

```

(5 PUNTOS) Crea o seguinte arquivo `octave.txt` (liñas separadas por “;”):

```
7 5 8; 4 2 6 1 6; 9 3; 1 8 5 4; 7
```

Escribe un programa en Octave chamado `exame.m` que lea tódolos números deste arquivo ao vector **x**. Sexa n a lonxitude de **x**, e $m=0$. Percorre **x** e para cada elemento x_i con $i = 1..n$ actualiza m como $m = ((i-1)m + x_i)/i$, rematando cando x_i sexa maior que $1.5m$. Borra os elementos x_j para $j = i+1..n$. Mostra por pantalla m e o novo vector **x**, nunha única liña. Define outro vector **y** como un ordeamento aleatorio dos elementos de **x**. Chama á función `func(·)`, que defina unha matriz **a** cadrada de orde n , sendo n é a lonxitude de **x**. Para calcular a_{ij} con $i, j = 1..n$ debes percorrer os elementos x_k de **x**, comezando en $k = \lfloor (i+2j)/3 \rfloor$ e facendo $k = 1$ se chegas a $k > n$, ata atopar un elemento x_k no vector **x** maior ou igual que o elemento y_k no vector **y**. O elemento a_{ij} será o número de elementos x_k percorridos. No programa principal: 1) resolve simbólicamente a ecuación $z^2 + \frac{1}{z} = 2$ e mostra por pantalla a súa segunda solución como número real con formato de ancho 10 e 5 decimais; e 2) representa gráficamente a matriz **a** como un mapa de calor usando o rango $1..n$ para os eixos X e Y e cunha barra de cor na dereita.

```

clear all;clc
%-----
function a=func(x,y,n)
    a=zeros(n);
    for i=1:n
        for j=1:n
            k=floor((i+2*j)/3);p=0;
            while x(k)<y(k)
                k=k+1;p=p+1;
                if k>n; k=1; end
            end
            a(i,j)=p;
        end
    end
end
%-----
m=0;n=0;nf='octave.txt';f=fopen(nf);
if f==-1; printf('erro fopen %s\n',nf);return; end
x=fscanf(f,'%d ');fclose(f);n=numel(x);
for i=1:n
    t=x(i);m=((i-1)*m+t)/i;
    if t>1.5*m; break; end
end
x(i+1:n)=[];n=numel(x);
disp('x=');disp(x);
printf('m=%g\n',m)
y=x(randperm(n));
a=func(x,y,n);
pkg load symbolic;syms z
t=solve(z^2+1/z-2,z);
printf('t=%10.5f\n',eval(t(2)))
[X,Y]=meshgrid(1:n);
figure(1);clf;colormap jet
contourf(X,Y,a);colorbar

```