

Exame de Programación Científica

1º Matemáticas, 25 xuño, 2026

NOTA: debes obter alomenos 1 punto en cada exercicio.

(5 PUNTOS) Escribe un programa `exame.f90` en Fortran que defina as constantes reais $p=0.2$, $a=0$ e $b=1$ e un vector $\mathbf{x}$ real con $n = 1 + \left\lfloor \frac{b-a}{p} \right\rfloor$ valores equiespaciados entre a e b , sendo $\lfloor \cdot \rfloor$ a parte enteira por defecto e p o paso ou distancia entre dous x_i consecutivos. Chama ao subprograma `almacena(·)`, do tipo e cos argumentos axeitados, que almacene no arquivo `fortran.txt` o vector $\mathbf{x}$ con formato real de ancho 5 e 2 decimais nunha soa liña. De volta no programa principal, crea un segundo vector $\mathbf{y}$ coa mesma lonxitude n , onde y_i con $i = 1..n$ sexa un elemento de $\mathbf{x}$ seleccionado aleatoriamente. Para facer isto, usa a función `rand()` que retorna un número real aleatorio en $[0,1]$, e a función `nint()` que da o enteiro máis cercano, para xerar un índice enteiro aleatorio entre 1 e n . Para evitar seleccionar varias veces un mesmo elemento de $\mathbf{x}$, usa un vector lóxico $\mathbf{u}$ de lonxitude n con elementos inicialmente falsos, poñendo despois u_i a certo cando se selecciona o elemento x_i . Finalmente, crea un vector $\mathbf{z}$ enteiro de lonxitude n tal que z_i para $i = 1..n$ sexa o número de elementos y_j do vector $\mathbf{y}$ anterior que hai que percorrer comezando en y_i ata que $y_j = x_i$. Se chegas ao final de $\mathbf{y}$, volve ao comezo. Mostra $\mathbf{x}$, $\mathbf{y}$ e $\mathbf{z}$ por pantalla, cada un nunha liña, co formato por defecto.

```
program exame
real,parameter :: p=0.2,a=0,b=1
real,allocatable :: x(:),y(:)
logical,allocatable :: u(:)
integer,allocatable :: z(:)
n=int(1+(b-a)/p)
allocate(x(n),y(n),u(n),z(n))
forall(i=1:n) x(i)=a+p*(i-1)
call almacena(x,n)
do i=1,n
  do
    j=nint((n-1)*rand()+1)
    if(.not.u(j)) then
      y(i)=x(j);u(j)=.true.
      exit
    end if
  end do
end do
do i=1,n
  j=i;v=x(i);k=1
  do
    j=j+1;k=k+1
    if(j>n) j=1
    if(y(j)==v) exit
  end do
  z(i)=k
end do
print ' ("x=", $) ';print *,x
print ' ("y=", $) ';print *,y
print ' ("z=", $) ';print *,z
deallocate(x,y,u,z)
end program exame
!-----
subroutine almacena(x,n)
real,intent(in) :: x(n)
integer,intent(in) :: n
open(1,file='fortran.txt')
```

```

do i=1,n
    write (1, '(f5.2, " ", $)') x(i)
end do
write (1, *); close(1)
end subroutine almacena

```

(5 PUNTOS) Crea o seguinte arquivo `octave.txt`, onde “;” separa liñas:

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

Escribe un programa en Octave chamado `exame.m` que lea o arquivo anterior liña a liña, almacenando nos vectores **x** e **y** os valores situados en posicións pares e impares da liña, respectivamente. Mostra **x** e **y**. Le por teclado e almacena na cadea de caracteres *s* a expresión analítica dunha función $f(x)$, introduce $x.^2.*\log(1+x.^2)-4$. Define *a* e *b* como o mínimo e máximo do vector **z** resultado de concatenar **x** e **y**. Define *n* como o mínimo das lonxitudes de **x** e de **y**. Chama á función `func(·)`, cos

argumentos axeitados, que calcule $t = \int_a^b f(x)dx$ empregando o comando de cálculo numérico que usa o método dos trapezios, para o cal debes converter *s* en función anónima. A función `func(·)` debe

calcular tamén a matriz **c** cadrada de orde *n* con elementos $c_{ij} = \sum_{k=\min(i,j)}^{\max(i,j)} x_k y_k$ para $i, j = 1..n$. De

volta no programa principal, mostra *t* con formato real de ancho 10 e 5 decimais. Representa a matriz **c** nun gráfico 3D cos eixos X e Y variando entre 1 e *n*, que na gráfica se deben chamar **Temperatura** e **Presión**, respectivamente. Engade unha barra de cores e pon *s* como título da gráfica.

```

clear all; clc
%-----
function [t, c]=func(s, a, b, x, y, n)
    u=linspace(a, b, n);
    f=str2func(sprintf('@(x) %s', s)); v=f(x);
    t=trapz(u, v); c=zeros(n);
    for i=1:n
        for j=1:n
            k=[i j]; m=min(k):max(k);
            c(i, j)=sum(x(m).*y(m));
        end
    end
end
%-----
nf='octave.txt'; f=fopen(nf);
if f==-1; printf('erro fopen %s', nf); return; end
x=[]; y=[];
while ~feof(f)
    l=fgetl(f); p=strsplit(l); z=str2double(p);
    x=[x z(2:2:end)]; y=[y z(1:2:end)];
end
fclose(f);
printf('x='); printf('%d ', x); printf('\n')
printf('y='); printf('%d ', y); printf('\n')
s=input('f(x)? ', 's');
z=[x y]; a=min(z); b=max(z);
n=min(numel(x), numel(y));
[t, c]=func(s, a, b, x, y, n);
printf('t=%10.5f\n', t)
clf; colormap jet
[X, Y]=meshgrid(1:n);
surf(X, Y, c); colorbar
xlabel('Temperatura'); ylabel('Presion'); title(s)

```