

Exame Informática, 1^o Matemáticas xaneiro, 2025, grupo E1

Debes obter 1 punto ou máis en: Maple, Fortran e no programa de Octave.

Apartado de Maple

1. (0.5 PUNTOS) Define a función $f(x, y, z) = \sin(xe^y \log(1 + z^2))$ e calcula $\frac{\partial^3}{\partial x \partial y \partial z} f\left(\frac{\pi}{3}, \frac{1}{2}, 1\right)$ con 4 decimais.

```
f := sin(x*exp(y)*log(z^2+1))
evalf(subs(x=Pi/3, y=1/2, z=1, diff(f, x, y, z)), 5)
```

2. (0.5 PUNTOS) Calcula $\sum_{n=1}^{\infty} \frac{n}{n^3 + 1}$.

```
sum(n/(n^3+1), n=1..infinity)
```

3. (0.5 PUNTOS) Representa gráficamente $xy^2 + \sin x \sin y = 2$.

```
with(plots): implicitplot(x*y^2+sin(x)*sin(y)-2, x=-10..10, y=-10..10)
```

4. (0.5 PUNTOS) Transforma o polinomio $3x^5 + x^2y^3 - 2xy^4 + y^4$ en $-2y^4x + y^4 + y^3x^2 + 3x^5$.

```
p := 3*x^5+x^2*y^3-2*x*y^4+y^4
sort(p, [y, x], plex)
```

Apartado de Fortran

(4 PUNTOS) Crea co editor Kate o seguinte arquivo `fortran1.txt`, onde o “;” separa liñas:

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

Escribe un programa en Fortran nomeado `exame1.f90` que lea o número n (neste caso $n=7$) na primeira liña do arquivo anterior e defina tres vectores enteiros \mathbf{x} , \mathbf{y} e \mathbf{z} , todos de lonxitude n . Logo debe ler e almacenar a primeira columna do arquivo no vector \mathbf{x} e a segunda columna no vector \mathbf{y} . Chama ao subprograma `subprog(·)` que defina o vector \mathbf{z} de lonxitude n onde o elemento z_i , con $i = 1..n$, se calcule do seguinte

modo: vai sumando o elemento x_j de \mathbf{x} comezando por $j = i$ ata que a suma supere o umbral $u_i = \sum_{k=1}^i y_k$.

O elemento z_i debe ser y_j , sendo j o índice do último elemento x_j de \mathbf{x} sumado. No programa principal, mostra o vector \mathbf{z} por pantalla nunha única liña.

```
program exame1
integer, allocatable :: x(:), y(:), z(:)
character(100) :: nf='fortran1.txt'
open(1, file=nf, status='old', err=1)
read(1,*) n
allocate(x(n), y(n), z(n))
do i=1, n
    read(1,*) x(i), y(i)
end do
close(1)
```

```

print *, 'x=', x
print *, 'y=', y
call subprog(x,y,z,n)
print *, 'z=', z
deallocate(x,y,z)
stop
1 print *, 'erro open ', nf
end program exame1
!-----
subroutine subprog(x,y,z,n)
integer, intent(in) :: x(n), y(n), n
integer, intent(out) :: z(n)
u=0
do i=1,n
    j=i; s=0; u=u+y(i)
    do while(s<=u)
        if(j>n) j=1
        s=s+x(j); j=j+1
    end do
    z(i)=y(j)
end do
end subroutine subprog

```

Apartado de Octave

1. (0.5 PUNTOS) Axusta os puntos $x=[1\ 2\ 3\ 4\ 5\ 6\ 7\ 8\ 9\ 10]$, $y=[9\ 8\ 7\ 5\ 6\ 4\ 5\ 7\ 9\ 10]$ a unha función potencial $y = ax^b$ e representa os puntos e a función.

```

x=1:10;y=[9 8 7 5 6 4 5 7 9 10];
p=polyfit(log(x),log(y),1);
a=exp(p(2));b=p(1);
x2=1:0.1:10;y2=polyval(p,x2);
plot(x,y,'o',x2,a*x2.^b,'r-')

```

2. (3.5 PUNTOS) Crea co Octave o arquivo `octave1.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 `exame1.m` que defina $n=5$ e lea número a número o arquivo anterior, almacenándoos no vector \mathbf{x} , mentres que a suma dos números lidos non supere o umbral $u = n^2$. Visualiza \mathbf{x} , a suma e o umbral. Chama á función `fun(·)` que calcule unha matriz \mathbf{a} e un vector \mathbf{y} . A matriz \mathbf{a} debe ser cadrada de orde m , sendo m a lonxitude de \mathbf{x} . Se $i \cdot j$ é par, debe ser $a_{ij} = \sum_{k=p}^q x_k x_{m-k+1}$ onde

$p = \min(i, j)$ e $q = \max(i, j)$. En caso contrario, $a_{ij} = j + \sum_{k=1}^i x_k$. O vector \mathbf{y} , de lonxitude m , debe ter

elementos $y_i = \sum_{j=1}^m a_{ij} x_j$ para $i = 1..m$. O programa principal debe visualizar \mathbf{a} (cada fila nunha liña) e \mathbf{y} (nunha única liña).

```

clear all;clc
%-----
function [a,y]=fun(x)
    m=numel(x);a=zeros(m);y=zeros(1,m);
    for i=1:m
        for j=1:m
            if rem(i*j,2)==0
                k=[i j];l=min(k):max(k);

```

```

        a(i,j)=x(1)*x(m-1+1)';
    else
        a(i,j)=sum(x(1:i))+j;
    end
end
y(i)=a(i,:)*x';
end
end
%-----
n=5;nf='octave1.txt';
f=fopen(nf);
if f==-1; error('fopen %s',nf); end
s=0;u=n^2;
while ~feof(f) && s<u
    t=fscanf(f,'%i',1);
    s=s+t;x=[x t];
end
fclose(f);
disp('x');disp(x)
printf('suma=%i umbral=%i\n',s,u)
[a,y]=fun(x);
disp('a');disp(a)
disp('y');disp(y)

```