

## Exame de Informática, 1º Matemáticas, xullo, 2010

NOTA: Debes acadar alomenos 1 punto en cada apartado.

**Apartado de Maple:** Escribe comandos en Maple que fagan o seguinte:

1. **(0.5 PUNTOS)** Define dous números enteiros  $n = 1234$  e  $m = 4567$ . Calcula o cociente e o resto da súa división enteira, determina se  $n$  é primo e descompón  $m$  en factores primos.

```
n = 1234; m = 4567; iquo(m, n); irem(m, n); isprime(n); ifactor(m)
```

2. **(0.5 PUNTOS)** Dados os polinomios de dúas variábeis  $p(x, y) = 3xy^2 - x^3 + yx^2 - y^3$  e  $q(x, y) = xy + 2x^2y - yx^3 + 6y^4x$ , calcula a forma simplificada da función racional  $p/q$  e a súa descomposición en fraccións parciais en  $x$ .

```
p := 3*x*y^2 - x^3 + y*x^2 - y^3; q := x*y + 2*y*x^2 - y*x^3 + 6*y^4*x;  
normal(p/q); convert(p/q, 'parfrac', x)
```

3. **(0.5 PUNTOS)** Representa gráficamente o lugar xeométrico dado pola expresión (onde  $t$  é o tempo)  
 $z = f(x, y, t) = \cos \left[ \cos \left( \frac{tx}{10} \right) \operatorname{sen} \left( \frac{ty}{10} \right) \right]$  para  $x, y \in [-10, 10], t \in [0, 10]$  con 50 fotogramas.

```
animate3d(cos(cos(t*x/10)*sin(t*y/10)), x = -10 .. 10, y = -10 .. 10, t = 1 .. 10, frames  
= 50)
```

4. **(0.5 PUNTOS)** Define a función de Maple  $f(x) = \frac{e^{-x^2/10} \sin(x)}{1 + e^{-10x}}$ . Como podemos aproximala nun entorno do punto  $x_0 = 0$  cun polinomio de orde 4?

```
f := x -> exp(-x^2/10)*sin(x)/(1+exp(-10*x)); taylor(f(x), x = 0, 4)
```

5. **(0.5 PUNTOS)** Define a expresión de Maple  $f(x, y) = x^2 - 3y \sin(y^2 + x)$ . Calcula a derivada parcial  $\frac{\partial^2 f}{\partial x \partial y}$  e a integral dobre  $\int_0^1 \int_{3-x}^{5+x} f(x, y) dy dx$

```
f := x^2-3*y*sin(y^2+x); diff(diff(f, x), y);  
int(x^2-3*y*sin(y^2+x), [y = 3-x .. 5+x, x = 0 .. 1])
```

**Apartado de Fortran: 4 PUNTOS** Escribe un programa en Fortran que declare unha matriz cadrada  $\mathbf{a}$  de orde 7, e un vector  $\mathbf{v}$  de 3 compoñentes (non uses reserva dinámica de memoria). O programa debe chamar a un subprograma `le_matriz(...)` (debes decidi-lo seu tipo e argumentos), que lea a matriz  $\mathbf{a}$  dende o arquivo `matriz.dat` e imprima a matriz por pantalla (cada fila nunha liña da terminal). Podes usar para o arquivo `matriz.dat` a seguinte matriz:

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

Logo, o programa principal debe ler por teclado 3 números enteiros (cada un cun valor entre 1 e 9) e almacenalos no vector  $\mathbf{v}$ . Podes probar co vector  $\mathbf{v} = (1, 2, 3)$ . O programa principal debe chamar entón a un novo subprograma `suma_numeros(...)` (debes decidi-lo seu tipo e argumentos), que busque na matriz  $\mathbf{a}$  ocorrencias de calquer compoñente do vector  $\mathbf{v}$ , e retorne ao programa principal a suma destas ocorrencias. Ademais, cando atope unha ocorrencia dunha compoñente de  $\mathbf{v}$ , debe poñer a 0 ese elemento da matriz  $\mathbf{a}$ . Antes de retornar, o subprograma debe imprimir-la nova matriz  $\mathbf{a}$  por pantalla (unha fila en cada liña da terminal). A suma retornada debe imprimirse dende o programa principal.

```

program ejercicio_fortran
integer, dimension(7, 7) :: a
integer, dimension(3) :: v

call le_matriz(a)

print *, "introduce v: "
read *, v

suma = suma_ocurrencias(v, a)
print *, "suma_ocurrencias= ", suma

stop
end program ejercicio_fortran

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
subroutine le_matriz(a)
integer, dimension(7, 7), intent(out) :: a

open(1, file = "matriz.dat", status = "old", err = 1)
print *, "matriz A: "
do i = 1, 7
    read (1, *) (a(i, j), j = 1, 7)
    print *, (a(i, j), j = 1, 7)
end do
close(1)
return
1 print *, "o arquivo matriz.dat non existe"
stop
end subroutine le_matriz

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
function suma_ocurrencias(v, a) result(suma)
integer, dimension(3), intent(in) :: v
integer, dimension(7, 7), intent(inout) :: a

suma = 0
do i = 1, 7
    do j = 1, 7
        do k = 1, 3
            if(a(i, j) == v(k)) then
                suma = suma + a(i, j)
                a(i, j) = 0
            end if
        end do
    end do
end do
print *, "matriz A nova: "
do i = 1, 7
    print *, (a(i, j), j = 1, 7)
end do

return
end function suma_ocurrencias

```

1. (0.5 PUNTOS) Calcula o valor mínimo da función  $f(x) = \frac{e^{-x^2/10} \sin(x)}{1 + e^{-10x}}$  e o valor de  $x$  no que se acada ese mínimo.

```
[xmin vmin]=fminbnd('exp(-x^2/10)*sin(x)/(1 + exp(-10*x))', -1, 6)
```

2. (0.5 PUNTOS) Representa gráficamente o lugar xeométrico  $z = f(x, y) = \cos \left[ \cos \left( \frac{xy}{50} \right) \operatorname{sen} \left( \frac{xy}{50} \right) \right]$  para  $x, y \in [-10, 10]$ .

```
[x y]=meshgrid(-10:0.1:10); z=cos(cos(x.*y/50).*sin(x.*y/50)); mesh(x, y, z)
```

3. (2 PUNTOS) Escribe un programa que cree unha matriz mágica de orde 7 e determine se algún dos seus elementos é simultaneamente o máximo da súa fila e o máximo da súa columna.

```
a = magic(7);
n = size(a, 1);
for i = 1:n
    for j = 1:n
        [max_fila i_max_fila] = max(a(i, :));
        [max_col i_max_col] = max(a(:, j));
        if i_max_fila == i && i_max_col == j
            fprintf('o elemento %i,%i con valor %g e o maximo da sua fila e da sua
                columna\n', i, j, a(i, j));
        end
    end
end
```

4. (0.5 PUNTOS) Crea un vector con 100 valores aleatorios no intervalo  $[-10, 10]$  e para mostrar por pantalla os elementos do vector superiores a 0 e os seus índices.

```
v = -10 + 20*random(1, 100); find(v > 0); v(find(v > 0))
```