

Informática. Exame de Teoría.
29 de novembro de 2006

Nome:

1	2	3	4	5	6	7	8	9	TOTAL

Responder brevemente ás seguintes cuestións:

1. (**2 PUNTOS**) Escribir unha subrutina en Fortran que reciba como argumentos unha matriz de reais, o seu número de filas e o seu número de columnas. A matriz debe calcular e proporcionar como resultado de saída a media dos elementos da matriz. A subrutina non debe imprimir nada por pantalla.

SOLUCIÓN

```
subroutine media_matriz(a, nf, nc, media)
  real, dimension(nf, nc), intent(in) :: a
  integer, intent(in) :: nf, nc
  real, intent(out) :: media

  media = 0
  do i = 1, nf
    do j = 1, nc
      media = media + a(i, j)
    end do
  end do
  media = media/(nf*nc)
  return
end subroutine media_matriz
```

2. (**1 PUNTO**) Éste é o pseudocódigo dun algoritmo de ordeamento que vimos na clase. Explica brevemente o seu funcionamento e o significado das variábeis `e`, `cambio`, `fin`, `aux`. Qué algoritmo é? Cómo opera sobre o vector { 3, 4, 2, 9, 1 } ?

```
e = n + 1
repetir
  e = enteira(e/2)
  repetir
    cambio = 0; i = 1
    repetir
      fin = 0
      se x(i) > x(i + e)
        aux = x(i); x(i) = x(i + 1); x(i + 1) = aux
        cambio = 1
      fin = 1
    fin = 1
  fin = 1
```

```

    fin_se
    se i + e == n
        fin = 1
    fin_se
    i = i + 1
    ata que fin == 1
    ata que cambio == 0
ata que e == 0
fin

```

SOLUCIÓN Trátase do algoritmo Shell, baseado na comparación de pares de números non consecutivos separados por un incremento e . Se o número esquerdo do par é maior co dereito, intercámbianse. O incremento vaise diminuindo conforme avanza o algoritmo, que por isto recibe o nome de ordeamento por incrementos decrecentes. `fin` é unha bandeira que vale 1 cando se alcanza o extremo esquerdo do vector de números. `cambio` é unha bandeira que vale 1 se se fixo algún cambio no percorrido actual do vector de números. `aux` serve como almacenamento temporal para os números intercambiados. Sobre o vector $\{ 3, 4, 2, 9, 1 \}$ o algoritmo Shell proposto opera así:

```

3 4 2 9 1, e = 3
3 4 2 9 1, comparo 3 e 9, non intercambia
3 1 2 9 4, comparo 4 e 1, intercambia, chego ao final, fin = 1, e = e/2 = 1
1 3 2 9 4, comparo 3 e 1, intercambia
1 2 3 9 4, comparo 3 e 2, intercambia
1 2 3 9 4, comparo 3 e 9, non intercambia
1 2 3 4 9, comparo 9 e 4, intercambia, chego ao final, fin = 1, e = 1,
fin do algoritmo

```

3. **(2 PUNTOS)** Escribir un algoritmo en diagrama de fluxo que lea a orde n e os coeficientes $a_0, \dots, a_n$ dun polinomio $p(x) = \sum_{i=0}^n a_i x^i$. O programa debe percorre-los divisores enteiros do termo independente a_0 e para cada un deles debe avaliar se é unha raíz do polinomio (é dicir, dado un divisor enteiro x^* de a_0 , avaliar se $p(x^*) = 0$). O programa debe imprimi-las raíces enteiras.

SOLUCIÓN Ver diagrama de fluxo adxunto

4. **(1 PUNTO)** O seguinte programa ten como obxectivo ler dous números por teclado e mostrar por pantalla a súa suma. Cres que ten algún erro? Xustifica a túa resposta.

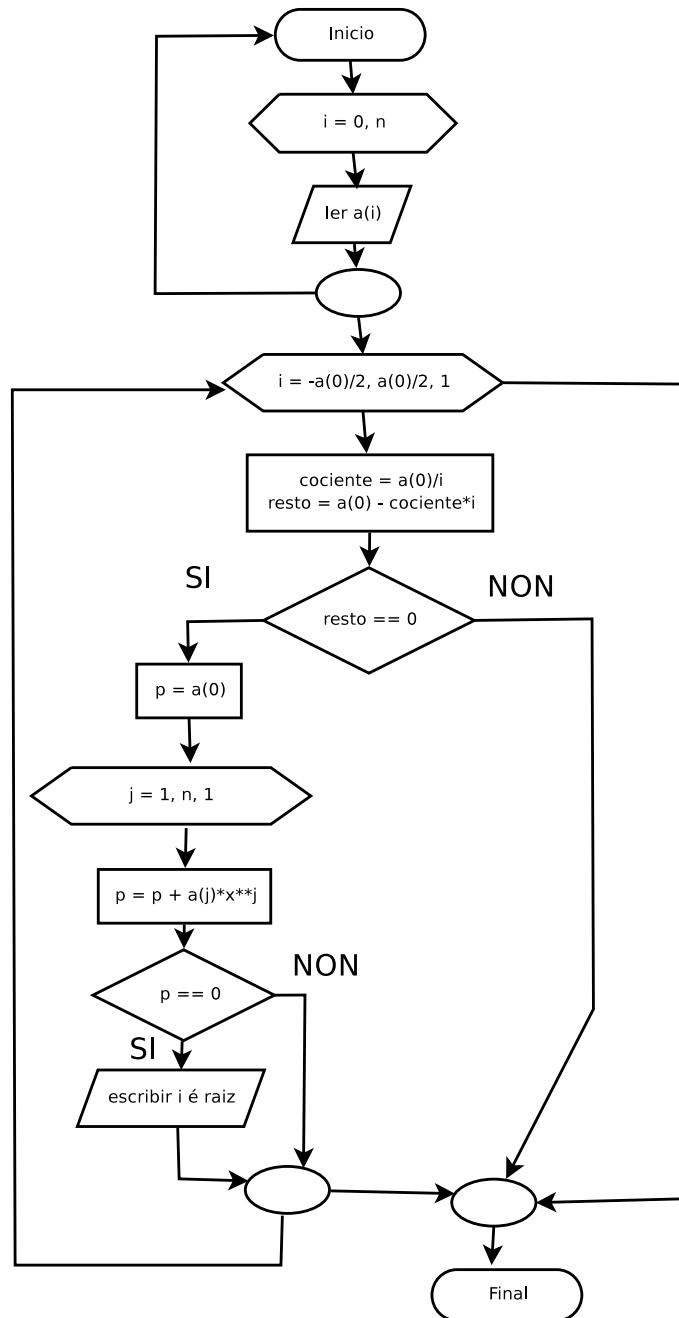
```

program suma.f
print *, "introduza x, y"
read *, x, y
real :: x
x + y
end program proba.f

```

SOLUCIÓN Hai varios erros:

- O nome do programa ten caracteres especiais (o punto) e non pode telos (erro de compilación)



- A sentencia **end program** tamén ten caracteres especiais e ten un nome distinto da sentencia **program**.
- Hai unha declaración de variábel (**x**) logo de sentencias que non son declaracións (erro de compilación). Asemade, a declaración non é necesaria.
- Hai unha expresión aritmética (**x + y**) cuxo resultado non se almacena en ningunha variábel nin se imprime por pantalla (erro de compilación)
- Non se imprime por pantalla a suma

5. (1 PUNTO) Supoñamos que $n = 2$, $m = 5$, $p = .true.$, usando declaración implícita de variábeis, e considerémo-la seguinte expresión en Fortran:

$(n + m)/2 == 1$.neqv. .not. p

De qué tipo é? Cál é o seu valor?

SOLUCIÓN A expresión é de tipo lóxico e o seu valor é `.false.`

```
(n + m)/2 == 1 .neqv. .not. p
2    5                      .true.
    7/2                      .false.
    3 == 1 .neqv. .false.
    .false. .neqv. .false.
    .false.
```

6. (0.5 PUNTOS) Explica cal é o erro (e de qué tipo é) no seguinte fragmento de sentencias en Fortran (supoñemos `y`, `z` adecuadamente inicializadas):

```
x = y + z - 2
x = -2
```

SOLUCIÓN Trátase dun erro lóxico producido por almacenar consecutivamente dous valores distintos na variábel `x`, de modo que o segundo valor sobrescribe ao primeiro, que se perde sen ser usado nunca. Polo tanto, eventualmente o programa proporcionará resultados erróneos, xa que perde información.

7. (1 PUNTO) Escribi-las sentencias en Fortran necesarias para:

- Declarar unha constante co valor $3.2 - i2.1$ (`i` é a unidade imaxinaria)
- Declarar unha variábel co valor $3,12 \times 10^{-123}$

SOLUCIÓN

- a) `complex, parameter :: c = (3.2, -2.1)`
b) `real(kind = 8) :: x = 3.12D-123`

8. (1 PUNTO) Escribi-las sentencias en Fortran necesarias para:

- Abrir un arquivo de disco chamado “`texto.txt`” xa existente e que se produza un erro se non existe.
- Ler do anterior arquivo dous valores reais con expoñente e 3 cifras decimais, e un enteiro de 6 cifras, pechando o arquivo cando se atopa a súa final.

SOLUCIÓN

a)

```
open(1, file = "texto.txt", status = "old", err = 1)
...
stop
1 print *, "erro en open"
stop
```

b)

```
read (1, fmt = 1, end = 2) x, y, n
1 format(2e9.3, i7)
2 close(1)
```

9. (0.5 PUNTOS) No contexto da programación en Fortran, qué é un erro por desbordamento de pila? Cándo se produce?

SOLUCIÓN É un erro de execución que se produce cando se supera o máximo número de chamadas a subprogramas realizadas, de modo que a pila de chamadas a subprogramas se desborda. Este erro prodúcese, por exemplo, cando un subprograma recursivo se chama a si mesmo indefinidamente, por exemplo debido a que non hai un caso non recursivo ou si que o hai, pero a condición non se cumpre nunca.