INFORMATICA

eiae
UPM

- Informática (1er cuatrimestre) – 40% nota final.
  - Sistema de evaluación.
    - 10% <u>Presentaciones en grupos de cuatro</u>.
    - Presentaciones cortas (10 minutos) sobre temas de teoría relacionados con la asignatura.
    - Crearé un Doodle con una lista de temas.
    - Os apuntáis y asignáis un número de grupo.
    - Después me escribís un correo con los nombres, dni y correo electrónico UPM de los integrantes.
    - (Lo volveré a explicar en clase cuando toque).

Presentaciones

- Links de doodle:
  - Grupo M1: https://doodle.com/poll/qc55m7ckuc366c2e
  - Grupo M3: https://doodle.com/poll/gb36fefwdapykvg5
  - Seguir el link correspondiente al grupo.

Presentaciones

- Links de doodle:
  - Grupo M1: https://doodle.com/poll/qc55m7ckuc366c2e
  - Grupo M3: https://doodle.com/poll/gb36fefwdapykvg5

  - El representante del grupo me envía un correo (g.rubio@upm.es).
  - Soy fulanito, con DNI tal, perteneciente al grupo MX y mi grupo está formado por bla, bla y bla con DNIs tal, tal y tal. Realizaremos la presentación: NOMBRE PRESENTACIÓN.

Presentaciones

- Las presentaciones empezarán el día 11 de Octubre.
- (Id pensando las parejas, a mediados del mes que viene os propongo el primer ejercicio entregable).

Hoy veremos…

- Sentencias condicionales:
  - Estructura `if … else if … endif`
  - Estructura `select case … case … default`
- Bucles iterativos:
  - Estructura `do … enddo`

eiae UPM

Antes:

- ¿Dudas práctica jueves pasado?

  …
- Comentarios:
  - División entera (4/3) = 1
  - Exponentes r**3.d0
  - Inicializar según el tipo:

    real*8,parameter :: pi=3.14159265358979d0

    real*8,parameter :: pi=acos(-1.d0)
  - Operador asignación *(a=b)*
  - Intercambiar variables

Antes:

Crea un nuevo proyecto (mismo procedimiento que antes) y usa el fichero principal2.f95 como programa principal. Corrige todos los errores y las advertencias que aparezcan cuando trates de compilarlo. Busca errores de programación no sintácticos.

```
program buscar errores

integer :: i,j,i_i

i=0.d0
j=1

real,parameter  ::pi=3.1415926535
real*8  :: x,y,X,z


y = 3
z =1.7d0

X = y*i

k = y*z

write(*,*) 'el valor de X es ',x
write(*,*) 'Pi con 10 cifras decimales vale: ',pi



end program
```

Antes:

r\EIAE\Informatica 15-16\1ºCuatrimestre\Codigos\Clase_3 - [primer_proyecto]

er Documento Proyecto Construir Herramientas Ayuda

ardar Guardar todo Revertir Cerrar Atrás Adelante Compilar Cons

estructuras.f90 | Geometria_Computacional.f90 | Aztec_file.f90 | boundary_lay

```fortran
program inicializar

real*8, parameter  :: pi = 3.14159265358979d0
real*8, parameter  :: pi_mal = 3.14159265358979
real*8, parameter  :: pi2 = acos(-1.d0)
real*8, parameter  :: pi2_mal =  acos(-1.)

 write(*,*) pi
 write(*,*) pi_mal
 write(*,*) pi2
 write(*,*) pi2_mal

 !write(*,*) (-2)**3d0  !Error
 write(*,*) (-2)**3

end program
```

C:\Windows\system32\cmd.exe

```
   3.1415926535897900
   3.1415927410125732
   3.1415926535897931
   3.1415927410125732
          -8
Presione una tecla para continuar . . .
```

eiae UPM

Recordamos estructura básica de un programa

```
program pepe

    implicit none

    ! Declaración de variables

    ! Inicialización de variables

    ! Cuerpo del programa


end program pepe
```

Operadores Numéricos

- Aritméticos

Suma	+
Resta	–
Producto	*
Cociente	/
Potenciación	**

- Relacionales

mayor que	>
menor que	<
mayor o igual que	>=
menor o igual que	<=
igual que	==
distinto que	/=

Se usan en expresiones lógicas cuya resultado es VERDAD o MENTIRA

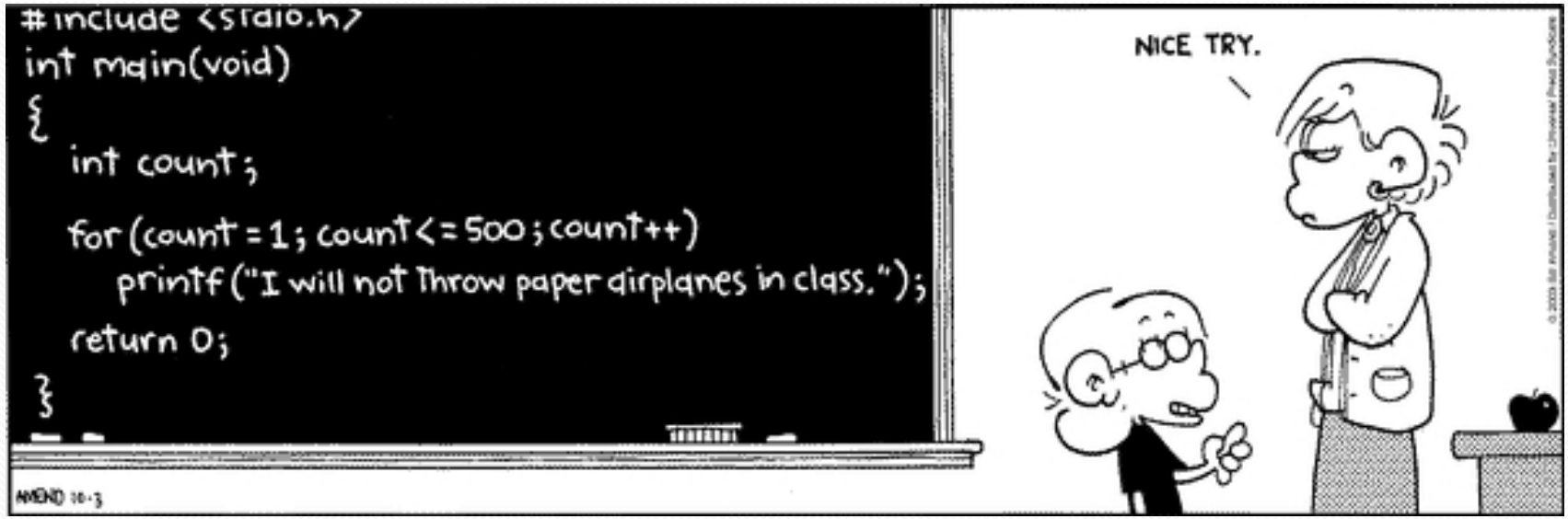
Problema para programar en casa

$$\text{Si } x < y \qquad \begin{cases} f(x,y) = sen^2(x) & x < -2 \\ f(x,y) = \sqrt{x^2+y^2} & -2 \le x < 2 \\ f(x,y) = \frac{x}{2y} & 2 \le x \le 4 \\ f(x,y) = 7x^{4/3} & x > 4 \end{cases}$$

$$\text{Si } x > y \quad e \quad y > -7 \qquad \begin{cases} f(x,y) = y - x & x < -2 \\ \begin{cases} f(x,y) = \frac{x}{y} & y \neq 0 \\ f(x,y) = 0 & y = 0 \end{cases} & -2 \le x < 2 \\ f(x,y) = |y| & x \ge 2 \end{cases}$$

$$\text{En el resto de los casos} \qquad f(x,y) = 7x^3 + 2x^2 - x + 5$$

Cuerpo del programa: Bucles iterados



Cuerpo del programa: Bucles iterados

Una estructura iterativa (*bucle*) *es aquella que* ***ejecuta repetidas*** **veces** un conjunto de sentencias (*rango del bucle*).

Existen dos tipos de estructuras iterativas:

```
do xxxxx

        sentencias

enddo
```

- Bucle controlado por un contador

Utiliza un contador para fijar a priori el número máximo de iteraciones a realizar

- Bucle controlado por una expresión lógica

Una expresión lógica controla la salida del bucle

Cuerpo del programa:

Bucle controlado por un contador

```
do i = Valor_inic, Valor_fin [, p]

        sentencia 1
        sentencia 2
        …
        sentencia n

enddo
```

- i es una variable, llamada *contador, de tipo* integer.

- `Valor_inic, Valor_fin, p` son constantes o variables de tipo integer, tales que `Valor_inic` es el valor inicial del contador, `Valor_fin` es el valor final del contador y p es el incremento del contador en cada iteración.

- El incremento `p` no puede ser cero.
- Si `p` =1, entonces no es necesario especificarlo.
- Si `p` <0, entonces se debe cumplir `Valor_inic > Valor_fin`

- NUNCA modificar la variable contador dentro de un bucle.
- Al salir del bucle el valor del contador es el primero fuera del rango al incrementarle p

eiae UPM

Ejemplos sencillos

```
program main
    integer :: i

    do i=1,3
        write(*,*) 'i vale:', i
    end do
    write(*,*) 'Fuera, i vale:', i

end program main
```

```
program main
    integer :: i

    do i=1,7,2
        write(*,*) 'i vale:', i
    end do
    write(*,*) 'Fuera, i vale:', i

end program main
```

Ejemplos sencillos

```
program main
    integer :: i

    do i=1,3
        write(*,*) 'i vale:', i
    end do
    write(*,*) 'Fuera, i vale:', i

end program main
```

```
C:\Program Files (x86)\Silverfrost\FTN95\Plato.exe
 i vale:            1
 i vale:            2
 i vale:            3
 Fuera, i vale:            4

Press RETURN to close window . . .
```

```
program main
    integer :: i

    do i=1,7,2
        write(*,*) 'i vale:', i
    end do
    write(*,*) 'Fuera, i vale:', i

end program main
```

```
C:\Program Files (x86)\Silverfrost\FTN95\Plato.exe
 i vale:            1
 i vale:            3
 i vale:            5
 i vale:            7
 Fuera, i vale:            9

Press RETURN to close window . . .
```

Ejemplo: 2 bucles anidados

```
program main
    integer :: i, j

    do i=3,8,2
        write(*,*) 'i vale:', i

        do j=3,5
            write(*,*) 'j vale:', j
        end do
        write(*,*) 'Fuera, j vale:', j

    end do
    write(*,*) 'Fuera, i vale:', i
    write(*,*) 'Fuera, j vale:', j

end program main
```

Ejemplo: 2 bucles anidados

```
program main
    integer :: i, j

    do i=3,8,2
        write(*,*) 'i vale:', i

        do j=3,5
            write(*,*) 'j vale:', j
        end do
        write(*,*) 'Fuera, j vale:', j

    end do
    write(*,*) 'Fuera, i vale:', i
    write(*,*) 'Fuera, j vale:', j

end program main
```

```
C:\Program Files (x86)\Silverfrost\FTN95\Plato.exe
 i vale:            3
     j vale:            3
     j vale:            4
     j vale:            5
         Fuera, j vale:            6
 i vale:            5
     j vale:            3
     j vale:            4
     j vale:            5
         Fuera, j vale:            6
 i vale:            7
     j vale:            3
     j vale:            4
     j vale:            5
         Fuera, j vale:            6
 Fuera, i vale:            9
 Fuera, j vale:            6

Press RETURN to close window . . .
```

Ejemplo: programa con sentencia if y bucle do

```
program sumatorio
    integer :: num
    integer :: suma
    integer :: i

    write(*,*) 'Introducir un numero > 0'
    read(*,*) num

    ! calculo de la suma 1 + 2 + ... + num



end program sumatorio
```

Ejemplo: programa con sentencia if y bucle do

```
program factorial
    integer :: num
    integer :: Fact
    integer :: i

    write(*,*) 'Introducir un numero >= 0'
    read(*,*) num

    ! calculo del factorial de num

end program factorial
```

Cuerpo del programa:

Bucle controlado por expresión lógica

Dos posibilidades:

OPCIONAL

```
do i = V_inic, V_fin [, p]

        sentencia 1
        …
        if (condicion logica) exit
        …
        sentencia n

enddo
```

Compilador standard:

```
do while (condicion logica)

        sentencia 1
        …
        sentencia n

enddo
```

Ejercicio:

Escribir un programa con las siguientes especificaciones:

- Pide que se introduzca por teclado una variable n de tipo integer.
- Cuando el valor introducido es un numero primo imprime un mensaje por pantalla y para el programa.

```
write(*,*) 'Escribe un numero natural'
read(*,*) n
```

Trabajo para casa

Escribe un programa que te calcule, con n un número entero que pidas por pantalla, los siguientes sumatorios

$$S_1 = \sum_{i=1}^{n} i^2$$

$$S_2 = \sum_{i=1}^{n} (i+1)^2$$

$$S_3 = \sum_{i=1}^{n} \sum_{j=1}^{m} \frac{i+j}{i}$$

$$S_4 = \sum_{i=1}^{n} \sum_{j=i}^{m} (i * j + i - j)$$

$$S_5 = \sum_{i=1}^{n} \sum_{j=i}^{n+m} (i - j)$$

$$S_6 = \sum_{i=1}^{n} \sum_{j=0}^{i} \frac{(i-j)^2}{3}$$