

- 1.– Crear en el disco duro dos carpetas llamada C y Fortran.  
Acceder a la página web

[http://caminos.udc.es/info/asignaturas/grado\\_itop/503](http://caminos.udc.es/info/asignaturas/grado_itop/503) ,

seguir la ruta

→ Software,  
→ 1\_Ejemplos\_C-Fortran,  
→ 1-1\_Hola/

y descargar el contenido de cada una de las dos carpetas existentes (C/ y Fortran/) en la correspondiente carpeta recién creada.

Se pide:

- a) Examinar, analizar y comparar los programas `hola.c` y `hola.f`.
- b) Ejecutar los dos programas `hola.exe` y comprobar que funcionan correctamente.
- c) Borrar los dos programas ejecutables `hola.exe`. Abrir una ventana en modo comando, regenerar el ejecutable de cada carpeta mediante las instrucciones

```
$ gcc hola.c -O2 -o hola.exe
```

```
$ gfortran hola.f -O2 -o hola.exe
```

y comprobar que funcionan correctamente.

- d) Borrar los dos programas ejecutables `hola.exe`. Abrir una ventana en modo comando, regenerar el ejecutable de cada carpeta mediante las instrucciones

```
$ gcc hola.c -O2 -c -o hola.o
```

```
$ gcc hola.o -O2 -o hola.exe
```

```
$ gfortran hola.f -O2 -c -o hola.o
```

```
$ gfortran hola.o -O2 -o hola.exe
```

y comprobar que funcionan correctamente.

- e) Analizar el contenido de los programas de comandos `make.bat`. Ejecutar estos programas de comandos en modo comando y en modo gráfico y comprobar que los ejecutables recién generados funcionan correctamente.

**Nota:** Los apartados b) y e) sólo tienen sentido si se trabaja en Windows, ya que tanto los ejecutables (archivos `*.exe`) como los programas de comandos (archivos `*.bat`) expuestos en la página web de la asignatura son propios de este tipo de sistemas.

---

**2.**— Crear en el disco duro una carpeta llamada Fortran.

Situarse dentro de la carpeta y crear los archivos de texto `prog.f`, `sub1.f`, `sub2.f` y `sub3.f`. Obsérvese el programa FORTRAN listado en las páginas siguientes.

Mediante una aplicación tipo Wordpad o un editor tipo SciTe, Emacs o Vim, se escribirá el programa principal en el archivo `prog.f` y se escribirán las subrutinas Generar, Alterar e Imprimir en los archivos `sub1.f`, `sub2.f` y `sub3.f` respectivamente.

Se pide:

a) Examinar, analizar y explicar el funcionamiento del programa principal y de cada una de las subrutinas.

b) Construir el programa ejecutable `prog.exe` mediante las instrucciones

```
$ gfortran prog.f sub1.f sub2.f sub3.f -O2 -o prog.exe
```

y comprobar que funciona correctamente.

c) Borrar el programa ejecutable `prog.exe` y reconstruirlo mediante las instrucciones

```
$ gfortran sub1.f -O2 -c -o sub1.o
```

```
$ gfortran sub2.f -O2 -c -o sub2.o
```

```
$ gfortran sub3.f -O2 -c -o sub3.o
```

```
$ gfortran prog.f -O2 -c -o prog.o
```

```
$ gfortran prog.o sub1.o sub2.o sub3.o -O2 -o prog.exe
```

```
$ del prog.o sub1.o sub2.o sub3.o
```

y comprobar que funciona correctamente.

d) Borrar el programa ejecutable `prog.exe` y reconstruirlo mediante las instrucciones

```
$ gfortran sub1.f -O2 -c -o sub1.o
```

```
$ ar r libsubs.a sub1.o
```

```
$ del sub1.o
```

```
$ gfortran sub2.f -O2 -c -o sub2.o
```

```
$ ar r libsubs.a sub2.o
```

```
$ del sub2.o
```

```
$ gfortran sub3.f -O2 -c -o sub3.o
```

```
$ ar r libsubs.a sub3.o
```

```
$ del sub3.o
```

```
$ gfortran prog.f -O2 -c -o prog.o
```

```
$ gfortran prog.o libsubs.a -O2 -o prog.exe
```

```
$ del prog.o
```

y comprobar que funciona correctamente.

---

C Programa de estilo primitivo

```
implicit real*8 (a-h,o-z)
parameter(MX=1000)
dimension v(MX)
```

```
1 write(6,100) ' Introduce el numero de componentes: '
100 format(/a,$)
    read(5,*) n
    if (n.lt.1.or.n.gt.MX) goto 1
```

```
call Generar(n,v)
call Alterar(n,v)
call Imprimir(n,v)
```

```
read(5,'( )')
call exit(0)
```

```
end
```

C  
C Subrutina *Generar*, de difícil comprensión y cuyo objetivo se desconoce  
C  
C Utiliza las funciones intrínsecas:  
C        *srand*(*iseed*)  
C                -> usa *iseed* (entero no negativo) como semilla de una  
C                secuencia de números pseudo-aleatorios  
C        *rand*( )  
C                -> devuelve un número pseudo-aleatorio  
C                (real\*4) en el rango 0.0 - 1.0  
C

---

```
Subroutine Generar(n,v)
implicit real*8 (a-h,o-z)
dimension v(n)

1  write(6,100) ' Introduce semilla (No. entero > 0): '
100  format(/a,$)
      read(5,*) iseed
      if (iseed.lt.0) goto 1

      call srand(iseed)
      i=1
2  v(i)=dbble(rand( ))
      i=i+1
      if (i.le.n) goto 2

      return
      end
```

C

C Subrutina Alterar, de difícil comprensión y cuyo objetivo se desconoce

C

```
Subroutine Alterar(n,v)
implicit real*8 (a-h,o-z)
dimension v(n)
```

```
    i=1
1   j=n
2   if (v(i).le.v(j)) goto 3
        vc =v(i)
        v(i)=v(j)
        v(j)=vc
3   j=j-1
    if(j.gt.i) goto 2
    i=i+1
if (i.lt.n) goto 1

return
end
```

C

C Subrutina Imprimir, de difícil comprensión y cuyo objetivo se desconoce

C

```
Subroutine Imprimir(n,v)
implicit real*8 (a-h,o-z)
dimension v(n)

write(6,100) ' RESULTADO:'
100 format(//a/)

i=1
1 write(6,200) i, v(i)
200 format(' v(',i5,') = ',f20.9)
i=i+1
if (i.le.n) goto 1

return
end
```