

**1.–** Crear en el disco duro una carpeta llamada C.

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

Mediante una aplicación tipo Wordpad o un editor tipo SciTe, Emacs o Vim, se escribirá la función `main()` en el archivo `prog.c` y se escribirán las funciones `Generar()`, `Alterar()` e `Imprimir()` en los archivos `sub1.c`, `sub2.c` y `sub3.c` 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

```
$ gcc prog.c sub1.c sub2.c sub3.c -O2 -o prog.exe
```

y comprobar que funciona correctamente.

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

```
$ gcc sub1.c -O2 -c -o sub1.o
$ gcc sub2.c -O2 -c -o sub2.o
$ gcc sub3.c -O2 -c -o sub3.o
$ gcc prog.c -O2 -c -o prog.o
$ gcc 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

```
$ gcc sub1.c -O2 -c -o sub1.o
$ ar r libsubs.a sub1.o
$ del sub1.o
$ gcc sub2.c -O2 -c -o sub2.o
$ ar r libsubs.a sub2.o
$ del sub2.o
$ gcc sub3.c -O2 -c -o sub3.o
$ ar r libsubs.a sub3.o
$ del sub3.o
$ gcc prog.c -O2 -c -o prog.o
$ gcc prog.o libsubs.a -O2 -o prog.exe
$ del prog.o
```

y comprobar que funciona correctamente.

---

```
/* Programa de estilo primitivo */

#include <stdlib.h>
#include <stdio.h>

#define MX 1000

int main(void)
{
    void Generar(int, double [ ]);
    void Alterar(int, double [ ]);
    void Imprimir(int, double [ ]);

    int n;
    double v[MX];

    LeeNComponentes:
        printf("\n Introduce el numero de componentes: ");
        scanf("%d", &n);
    if (n < 1 || n > MX) goto LeeNComponentes;

    Generar(n, v);
    Alterar(n, v);
    Imprimir(n, v);

    fflush(stdin); getchar( );
    exit(0);
}
```

/\*

---

Funcion *Generar*, de dificil comprension y cuyo objetivo se desconoce

Utiliza las funciones de la libreria standard:

`void srand(unsigned int iseed)`  
    -> usa *iseed* como semilla de una secuencia  
        de numeros pseudo-aleatorios

`int rand(void)`  
    -> devuelve un numero pseudo-aleatorio  
        (entero) en el rango 0-RAND\_MAX

\*/

```
#include <stdlib.h> /* Declara funciones srand( ), rand( ) */  
#include <stdio.h>
```

```
void Generar(int n, double v[ ])  
{  
    int iseed,  
        i;  
  
    LeeSemilla:  
        printf("\n Introduce semilla (No. entero > 0): ");  
        scanf("%d", &iseed);  
    if (iseed < 0 ) goto LeeSemilla;  
  
    srand(iseed);  
    i = 0;  
    Generar:  
        v[i] = (double) ((float) ((double) rand( ) / (double) RAND_MAX));  
        i++;  
    if (i < n) goto Generar;  
}
```

```
/*  
_____  
Funcion Alterar, de dificil comprension y cuyo objetivo se desconoce  
*/
```

```
#include <stdlib.h>
```

```
void Alterar(int n, double v[ ])
```

```
{  
    int i, j;  
    double vc;  
  
    i = 0;  
    BucleExterno:  
        j = n - 1;  
        BucleInterno:  
            if (v[i] <= v [j]) goto Seguir;  
                vc = v[i];  
                v[i] = v[j];  
                v[j] = vc;  
            Seguir:  
                j--;  
            if (j > i) goto BucleInterno;  
            i++;  
        if (i < n - 1) goto BucleExterno;  
}
```

```
/*  
_____  
Funcion Imprimir, de dificil comprension y cuyo objetivo se desconoce  
*/
```

```
#include <stdlib.h>
```

```
#include <stdio.h>
```

```
void Imprimir(int n, double v[ ])
```

```
{
```

```
    int i;
```

```
    printf("\n\n RESULTADO:\n\n");
```

```
    i = 0;
```

```
    Escribir:
```

```
        printf("  v[%5d] = %20.9lf\n", i, v[i]);
```

```
        i++;
```

```
    if (i < n) goto Escribir;
```

```
}
```