

# Tema F

## Ejercicio 1

Considerar la siguiente asignación múltiple:

```
var x, y : Int;  
var z : Bool;  
{Pre: x = X, y = Y, z = Z, Y ≠ 0}  
x, y, z := x*x + y*y, x mod y, x < y  
{Post: x = X*X + Y*Y, y = X mod Y, z = X < Y}
```

Escribir un programa en lenguaje C equivalente usando asignaciones simples teniendo en cuenta:

- Se deben verificar la pre y post condición usando la función `assert()`.
- Los valores iniciales de `x`, `y` deben obtenerse del usuario usando la función `pedirEntero()` definida en el *Proyecto 3*
- El valor inicial de `z` debe obtenerse del usuario usando la función `pedirBooleano()` definida en el *Proyecto 3*
- Los valores finales de `x`, `y` deben mostrarse por pantalla usando la función `imprimeEntero()` definida en el *Proyecto 3*.
- El valor final de `z` debe mostrarse por pantalla usando la función `imprimeBooleano()` definida en el *Proyecto 3*.

## Ejercicio 2

Programar la función:

```
int maximo_negativo(int a[], int tam);
```

que dado un arreglo `a[]` con `tam` elementos devuelve el elemento negativo más grande de `a[]`. Por ejemplo:

a[]	tam	resultado
[8, -100, 2, -5, 3]	5	-5
[3, 8, -1, -3, 5]	5	-1

Si en el arreglo `a[]` no hubiesen elementos negativos, la función debe devolver el neutro de la operación *máximo* para el tipo `int` (usar `<limits.h>`)

Cabe aclarar que `maximo_negativo` no debe mostrar ningún mensaje por pantalla ni pedir valores al usuario.

En la función `main` se debe solicitar al usuario ingresar un arreglo de longitud `N` (definir a `N` como una constante, el usuario no debe elegir el tamaño del arreglo) y luego mostrar el resultado de la función `maximo_negativo`.

## Ejercicio 3

Hacer un programa que, dado un arreglo `a[]` y su tamaño `tam` obtenga el mínimo y máximo elemento del arreglo `a[]` junto con sus correspondientes posiciones en el arreglo. Para ello programar la siguiente función:

```
struct maxmin_t cotas(int a[], int tam);
```

donde la estructura `struct maxmin_t` se define de la siguiente manera:

```
struct maxmin_t {  
    int minimo;  
    int indice_min;  
    int maximo;  
    int indice_max;  
}
```

La función toma un arreglo `a[]` y su tamaño `tam` devolviendo una estructura con cuatro enteros que contienen: el mínimo elemento (`minimo`), su posición en el arreglo (`indice_min`), el máximo elemento (`maximo`) y su posición en el arreglo (`indice_max`).

La función `cotas` debe implementarse con un único ciclo y **no debe mostrar mensajes por pantalla ni pedir valores al usuario**.

En la función `main` se debe solicitar al usuario ingresar un arreglo de longitud `N` (definir a `N` como una constante, el usuario no debe elegir el tamaño del arreglo). Finalmente mostrar el resultado de la función `cotas`

## Ejercicio 4\*

Hacer un programa que cuente la cantidad de personas que miden más y las que miden menos de una cantidad `c` que viene dada por un parámetro.

Para ello definimos primero una estructura `struct persona` que contiene el DNI de una persona y su altura, de la siguiente manera:

```
struct persona {
    int dni;
    float altura;
}
```

Luego se debe programar la función

```
struct alturas_t contar_altos_y_bajos(struct persona a[], int tam, float alt);
```

donde la estructura `struct alturas_t` se define de la siguiente manera:

```
struct alturas_t {
    int n_altos;
    int n_bajos;
}
```

La función toma un arreglo `a[]`, su tamaño `tam` y una altura `alt` para poder comparar.

La función devuelve una estructura con los dos enteros que respectivamente indican la cantidad de personas existentes en el arreglo `a[]` que son más altas que `alt`, y en el segundo entero la cantidad de personas que son más bajas.

La función `contar_altos_y_bajos` debe implementarse con un único ciclo y **no debe mostrar mensajes** por pantalla **ni pedir valores al usuario**.

En la función `main` se debe solicitar al usuario ingresar un arreglo de longitud `N` (definir a `N` como una constante, el usuario no debe elegir el tamaño del arreglo), también se debe pedir la altura para comparar, y por último, se debe mostrar el resultado de la función por pantalla.