

Recuperatorio: Tema C

Ejercicio 1

Considerar el siguiente asignación múltiple de variables:

```
var x, y, z : Int;  
{Pre: x = X, y = Y, z = Z, x ≠ y}  
x, y, z := y, x, x - y  
{Post: x = Y, y = X, z ≠ 0}
```

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`, `z` deben obtenerse del usuario desde el `main` con la función `pedirEntero` definida en el Proyecto 3.
- Los valores finales de `x`, `y`, `z` deben mostrarse por pantalla desde el `main` con la función `imprimirEntero` definida en el Proyecto 3.

Ejercicio 2

Programar la función:

```
bool todos_pares(int a[], int tam, int pos1, int pos2);
```

que dado un arreglo de enteros de `tam` elementos, devuelva `true` si entre los índices `pos1` y `pos2` todos los elementos de `a[]` son pares. En caso contrario devolverá `false`.

a[]	tam	pos1	pos2	resultado
[3, 2, 6, 30, 7]	5	1	3	true
[5, 16, 6, 20, 1]	5	0	4	false

- Chequear con `assert` que `pos1` y `pos2` estén entre las posiciones válidas del arreglo.
- Verificar con `assert` que `pos1` es menor a `pos2`.

Desde el `main` se debe solicitar al usuario 2 enteros: `pos1` y `pos2` con la función `pedirEntero()` del Proyecto 3.

Solicitar los números del arreglo y llamar a la función `todos_pares` e imprimir su resultado.

El tamaño del arreglo debe ser una constante, que no debe ingresar el usuario.

Ejercicio 3

Hacer una función que devuelva 5 números primos del arreglo en la estructura cinco. Si no hay 5 primos completar los elementos de la estructura con `-1`. Para ello programar la siguiente función:

```
struct cinco_t cinco_primos(int a[], int tam);
```

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

```
struct cinco_t {  
    int primero;  
    int segundo;  
    int tercero;  
    int cuarto;  
    int quinto;  
}
```

La función toma un arreglo `a[]` y su tamaño `tam`, devolviendo una estructura con los cinco números que son primos en el arreglo `a[]`, o `-1` cuando no haya. La función `cinco_primos` 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) y luego se debe mostrar el resultado de la función por pantalla.

Puede usar la función `es_primo` del Proyecto 3.