

# Parcial 2 - Algoritmos I Taller: Tema A

## Ejercicio 1

Considerar la siguiente asignación múltiple:

```
var r, s : Int;  
{Pre: r = R, s = S, S > R}  
r, s := s - r, r + s  
{Pos: r = S - R, s = R + S}
```

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

- Se deben verificar las pre y post condiciones usando la función `assert()`.
- Los valores iniciales de `r` y `s` deben ser ingresados por el usuario
- Los valores finales de `r` y `s` deben mostrarse por pantalla.

## Ejercicio 2

Programar la función:

```
int suma_mayores(int a[], int tam, int k);
```

que dado un arreglo `a[]` con `tam` elementos devuelve la suma de los valores de `a[]` que son estrictamente mayores al elemento de la posición `k` del arreglo. Por ejemplo:

a[]	tam	k	resultado	Comentario
[3, -5, 1, 9, 7]	5	2	<b>19</b>	Se suman sólo los elementos 3, 9 y 7 ya que son los únicos mayores a 1, que es el elemento ubicado en la posición k=2
[3, -5, 1, 9, 7]	5	0	<b>16</b>	Se suman sólo los elementos 9 y 7 ya que son los únicos mayores a 3, que es el elemento ubicado en la posición k=0
[3, -5, 1, 9, 7]	5	1	<b>20</b>	Se suman los elementos 3, 1, 9 y 7 ya que son los elementos mayores a -5, que es el elemento ubicado en la posición k=1
[3, -5, 2, 9, 7]	5	3	<b>0</b>	No se suma ningún número ya que no hay elementos mayores que 9, que es el elemento ubicado en la posición k=3

Cabe aclarar que `suma_mayores` 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 se debe pedir el índice `k` (verificar con `assert` que `k` es un número  $\geq 0$  y  $< N$ ) y finalmente mostrar el resultado de la función `suma_mayores`.

## Ejercicio 3

Hacer un programa que cuente la cantidad de elementos pares y la cantidad de elementos impares de un arreglo. Para ello programar la siguiente función

```
struct paridad_t contar_paridad(int a[], int tam);
```

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

```
struct paridad_t {  
    int n_pares;  
    int n_impares;  
}
```

La función toma un arreglo `a[]` y su tamaño `tam`, y devuelve una estructura con dos enteros que respectivamente indican cuántos elementos pares y cuántos impares hay en `a[]`. La función `contar_paridad` 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.

## Ejercicio 4\*

Hacer un programa que dado un arreglo de personas calcule la altura media, la mínima y la máxima. Para ello programar la siguiente función:

```
struct stats_t calcular_estadisticas(struct persona_t a[], int tam);
```

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

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

y la estructura `struct stats_t` se define como:

```
struct stats_t {  
    float altura_media;  
    float altura_minima;  
    float altura_maxima;  
};
```

La función toma un arreglo `a[]` con `tam` elementos de tipo `struct persona_t` y devuelve una estructura con tres números que respectivamente indican la altura promedio, la altura mínima y la altura máxima de las personas que hay en `a[]`. La función `calcular_estadisticas` 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 elementos de tipo `struct persona_t` de longitud `N` (definir a `N` como una constante, el usuario no debe elegir el tamaño del arreglo). Para ello solicitar por cada elemento del arreglo un valor entero y luego un valor flotante (usar `%f`). Se puede modificar la función `pedirArreglo()` para facilitar la entrada de datos. Luego se debe mostrar el resultado de la función `calcular_estadisticas` por pantalla.