

Parcial 2 - Algoritmos I Taller: Tema B

Ejercicio 1

Considerar la siguiente asignación múltiple:

```
var r, s : Int;  
{Pre: r = R, s = S, S > R}  
s, r := 2 * r - 2 * s, 2 * r + s  
{Pos: s = 2 * R - 2 * S, r = 2 * 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_multiplos(int a[], int tam, int k);
```

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

<code>a[]</code>	<code>tam</code>	<code>k</code>	resultado	Comentario
[3, -5, 2, 4, 7]	5	2	6	Se suman sólo los elementos 2 y 4 ya que son los únicos múltiplos de 2, que es el elemento ubicado en la posición <code>k=2</code>
[3, -5, 1, 9, 7]	5	0	12	Se suman sólo los elementos 3 y 9 ya que son los únicos múltiplos de 3, que es el elemento ubicado en la posición <code>k=0</code>
[3, 5, 1, 9, 7]	5	1	5	No se suma ningún elemento ya que ninguno es múltiplo de 5, que es el elemento ubicado en la posición <code>k=1</code>
[7, 14, 21, 7, 7]	5	3	56	Se suman todos los elementos ya que son múltiplos de 7 que es el elemento ubicado en la posición <code>k=3</code>

Cabe aclarar que `suma_multiplos` 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 ≥ 0 y $< N$) y finalmente mostrar el resultado de la función `suma_multiplos`.

Ejercicio 3

Hacer un programa que cuente la cantidad de elementos múltiplos de 2 y la cantidad de elementos múltiplos de 3 de un arreglo de enteros. Para ello programar la siguiente función:

```
struct cantidad_t contar_multiplos(int a[], int tam);
```

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

```
struct cantidad_t {  
    int n_multiplos_dos;  
    int n_multiplos_tres;  
}
```

La función toma un arreglo `a[]` y su tamaño `tam`, y devuelve una estructura con dos enteros que respectivamente indican cuántos elementos múltiplos de 2 y cuantos elementos múltiplos de 3 hay en `a[]`. La función `contar_multiplos` 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 stock de productos calcule la cantidad media, la mínima y la máxima. Para ello programar la siguiente función:

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

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

```
struct producto_t {
    int codigo;
    int cantidad;
};
```

y la estructura `struct stats_t` se define como:

```
struct stats_t {
    float cantidad_media;
    int cantidad_minima;
    int cantidad_maxima;
}
```

La función toma un arreglo `a[]` con `tam` elementos de tipo `struct producto_t` y devuelve una estructura con tres números que respectivamente indican la cantidad promedio, la cantidad mínima y la cantidad máxima de los productos 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 producto_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 otro valor entero. 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.