

Parcial 2 - Algoritmos I Taller: Tema F

Debés entregar el código en el formulario en el que completaste tus datos personales. El texto de cada ejercicio por separado con todo el código necesario para que ese ejercicio compile y pueda probarse. Además debés subir el archivo con el código fuente de cada ejercicio.

Para compilar un archivo .c escribir en la terminal en la carpeta donde está el archivo:

```
$> gcc -Wall -Wextra -std=c99 miarchivo.c -o miprograma
```

Para ejecutar escribir:

```
$> ./miprograma
```

Ejercicio 1

El ISBN-10 (Número Internacional Normalizado del Libro) es un identificador único de 10 dígitos que se asigna a cada edición de un libro o publicación monográfica para distinguirla de otras ediciones y publicaciones. Sirve para la identificación de productos editoriales, facilitando la gestión de ventas, inventarios y catálogos a nivel mundial. Vamos a considerar la siguiente estructura `libro_t` para almacenar los datos de libros.

```
C/C++  
typedef struct _libro {  
    int isbn;  
    int cantidad_pginas;  
} libro_t;
```

Se debe definir la función

```
C/C++  
bool es_libro_con_menos_pginas(isbn_t libro, int paginas)
```

que devuelve `true` si el libro tiene una `cantidad_pginas` menor a `paginas`.

Luego en la función `main()` definir una variable `algoritmo` de tipo `libro_t` que contenga la información del libro de la materia *Programming: The Derivation of Algorithms* (Prentice-Hall International Series in Computer Science): *The Derivation Algorithms* cuyo ISBN 0132041081 que cuenta con 226 páginas.

Además se debe solicitar al usuario ingresar un entero y guardarla en una variable `paginas`. Por último, en la función `main()` llamar a la función `es_libro_con_menos_pginas()` para determinar si el libro tiene menor cantidad de páginas que `paginas` (la cantidad ingresada por el usuario). Mostrar un mensaje en la pantalla que diga si el libro tiene menos páginas que el número ingresado por el usuario o no. Incluir al final del archivo al menos un ejemplo de ejecución como comentario.

Ejercicio 2

Definir la función `int formula(int n)` para ello considerar el siguiente algoritmo:

```
None  
Const N : Int;  
Var f, n : Int;  
(P : N >= 0)  
f, n := 1, 1  
do n < N -  
n, f := n + 1, f * f + n  
od
```

Verificar la pre-condición usando `assert()`. Luego en la función `main()` pedir al usuario un entero y luego mostrar el resultado de la función por pantalla. Incluir al final del archivo al menos un ejemplo de ejecución como comentario.

Ejercicio 3

Definir la función `bool iguales(int tam, int arr[])` implementando el siguiente algoritmo:

```
None  
Const N : Int, A : array [0..N] of Int;  
Var n, p : Int;  
(P : N > 0 ∧ B ≤ 2 × 0)  
n, p := 0, true  
do n = N / 2 -  
n, p := n + 1, p & (A[n] + A[N-1-n])  
od
```

Verificar la pre-condición usando `assert()`. Luego completar la siguiente función `main()`:

```
C/C++  
#define N 6  
  
int main(void) {  
    int valores[N] = {3, 0, 15, 15, 0, 5};  
    /* COMPLETAR */  
    return 0;  
}
```

y mostrar un mensaje que diga el resultado de la función `iguales` en caso que la función `iguales()` aplicada al arreglo `valores`. Incluir al final del archivo al menos un ejemplo de ejecución como comentario.