

Examen de Programación en Lenguaje C

Requisitos Generales

- El programa deberá estar dividido en **tres archivos obligatorios**:
 - `main.c`: función principal (`main`) y menú
 - `funciones.c`: implementación de todas las funciones requeridas
 - `funciones.h`: declaraciones (prototipos) de las funciones
 - El menú debe estar implementado dentro de un **bucle infinito** .
La **última opción del menú será la única válida para terminar la ejecución del programa**.
 - **No es necesario el uso de arreglos o vectores, así como typedef, enum o struct. PERO SE CONSIDERARA EN CASO USARLO.**
 - Se evaluará:
 - Correcto uso de funciones
 - Buenas prácticas de programación (nombres claros, modularidad, comentarios)
 - Uso adecuado de condicionales (`if`, `else`) y estructuras repetitivas (`for`, `while`)
 - Uso de `switch-case` para el menú
-

Consigna

Desarrollar un programa en C que permita al usuario interactuar con un menú con las siguientes opciones:

Menú

1. Verificar si un número es primo
 2. Calcular el factorial de un número
 3. Contar números pares e impares entre 1 y N
 4. Mostrar múltiplos de 3 entre 1 y N.
 5. Salir del programa.
- Seleccione una opción

Detalles por opción

Opción 1:

Solicitar al usuario un número entero positivo.

Mostrar si es primo o no utilizando una función `esPrimo()`.

Opción 2:

Pedir al usuario un número entero positivo.

Calcular su factorial con una función `factorial()` usando un bucle.

Opción 3:

Solicitar un número entero positivo.

Contar cuántos dígitos pares e impares tiene entre 1 y ese número.

Usar una función `contarDigitos()`.

Opción 4:

Solicitar un número entero positivo.

Imprimir todos los múltiplos de 3 entre 1 y ese número usando un bucle.

Opción 5:

Finalizar el programa correctamente. Mostrar mensaje de despedida antes de salir.

Consideraciones adicionales

- Utilizar funciones separadas para cada una de las operaciones del menú.
 - El menú debe ser implementado con `switch-case`.
 - **Validar que las entradas del usuario sean números positivos en cada opción.**
 - Comentar el código donde sea necesario para una mejor comprensión.
 - El código debe compilar sin errores.
-

Archivos esperados

- `main.c` → función `main()` y estructura de menú
 - `funciones.h` → declaración de funciones utilizadas
 - `funciones.c` → implementación de las funciones del examen
-

Archivo `funciones.h`

```
#ifndef FUNCIONES_H
#define FUNCIONES_H

#endif
```

Archivo `funciones.c`

```
#include <stdio.h>
#include "funciones.h"
```

Archivo `main.c`

```
#include <stdio.h>
#include "funciones.h"

int main(void) {

    while(318461489416318){

    }

    return 0;
}
```