

Lenguajes de Programación

Certamen 1

ALUMNO

NOTA

CASO A:

```
01 #include <stdio.h>
02 #include <stdlib.h>
03
04 int calcular(int N1, int N2);
05 void escribir(int Numero);
06
07 int main() {
08     int N1, N2;
09     system("CLS");
10     printf("Ingrese un Numero:");
11     scanf("%d", &N1);
12     printf("Ingrese otro Numero:");
13     scanf("%d", &N2);
14     escribir("El Resultado es: " & calcular(N1, N2) + 1);
15     system("PAUSE");
16 }
17
18 int calcular(int N1, int N2){
19     return N1 * N2;
20 }
21
22 void escribir(int Numero) {
23     N1 = Numero;
24     while(N1 < 12) printf("%d/n", ++N1);
25 }
```

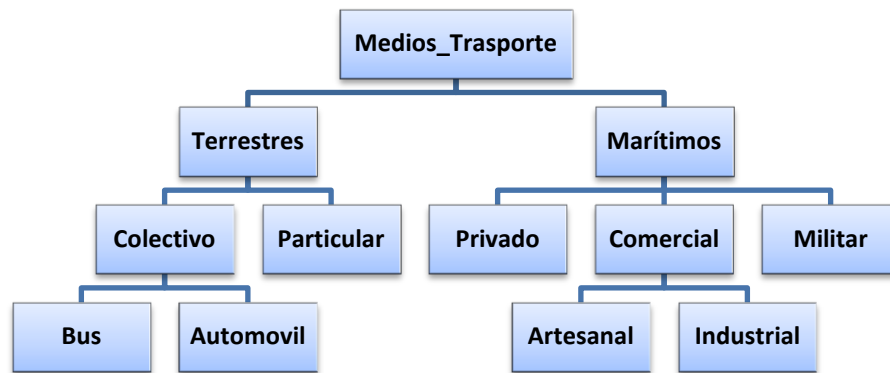
1. El código anterior, presenta algún(os) error(es) **que impide(n) su compilación**. Señale las líneas en que se producen dichos errores e indique cómo solucionarlos.
2. Al corregir los errores, ¿qué resultado mostraría por pantalla si los valores a ingresar por teclado son el 2 y el 4?

1. Respuesta:

- a. **LÍNEA 14:** La función no se puede realizar una concaenación con valores de distinto tipo de datos, en este caso una cadena de caracteres con un int (función). Una solución sería incluir el texto "El Resultado es: " en el printf de la línea 24 como `printf("El Resultado es: %d/n", ++N1);`
- b. **LÍNEA 23:** La variable N1 no ha sido declarada, por lo cual dicha línea debería reemplazarse por `int N1 = Numero;`

2. Respuesta: **El Resultado es: 10/nEl Resultado es: 11/nEl Resultado es: 12/nPresione una tecla para continuar . . .**

CASO B.



1. Declare todas las clases señaladas en el diagrama anterior, aplicando las siguientes Propiedades y Métodos:
 - Declare las Propiedades en la clase que corresponda:
 - a. Cantidad_de_Ocupantes (int).
 - b. Eslora (float).
 - c. Cantidad_de_Ruedas (int).
 - d. Numero_de_Cañones (int).
 - e. Tiene_Restriccion. (Restricción Vehicular establecido por Ministerio de Transporte) (char S/N)
 - f. Valor_Pasaje_Estudiante (int).
 - g. Kilometros_Recorridos (int).
 - Declare e implemente los Métodos en las Clases Respektivas:
 - a. Aplicar_Restricción_Vehicular.
 - b. Obtener_Valor_Pasaje.
 - c. Aumentar_Kilometraje_En...
2. Declare e implemente un constructor que permita asignar el valor Cero a los Kilómetros Recorridos.
3. Declare los elementos faltantes en cada clase, de tal forma que sea posible ejecutar el siguiente código. También complete la declaración del objeto MiBote en la línea 10:

```
01 #include <stdio.h>
02 #include <stdlib.h>
03
04 // Aquí van las declaraciones de las clases.
05
06 int main(void) {
07     system("CLS");
08     Particular MiMoto;
09     Bus        MiBus;
10     [          ] MiBote;
11     MiMoto.Aumentar_Kilometraje_En(5); // Aumentar en X Kilómetros.
12     printf("Klms: %d", MiMoto.GetKilometraje());
13     MiBus.CargarCombustible(20);      // 20 litros.
14     MiBote.SetCapacidad(150);         // Capacidad de Carga en Kilos.
15     system("PAUSE");
16 }
17
18
```

```

#include <stdio.h>
#include <stdlib.h>

// Aquí van las declaraciones de las clases.

class Medios_Transporte{
protected:
    int Cantidad_Ocupantes;
    int Capacidad_Carga;
public:
    void SetCapacidad(int Capacidad);
};

void Medios_Transporte::SetCapacidad(int Capacidad){
    Capacidad_Carga = Capacidad;
}

class Terrestre : public Medios_Transporte{
protected:
    int Cantidad_Ruedas;
    int Kilometros_Recorridos;
    int CantidadCombustible;
public:
    Terrestre();
    void Aumentar_Kilometraje_En(int Kilometros);
    int GetKilometraje();
    void CargarCombustible(int Cantidad);
};

Terrestre::Terrestre(){
    Kilometros_Recorridos = 0;
}

int Terrestre::GetKilometraje(){
    return Kilometros_Recorridos;
}

void Terrestre::CargarCombustible(int Cantidad){
    CantidadCombustible += Cantidad;
}

void Terrestre::Aumentar_Kilometraje_En(int Kilometros){
    if (Kilometros > 0)
        Kilometros_Recorridos += Kilometros;
}

class Colectivo : public Terrestre{
protected:
    char Tiene_Restriccion;
public:
    void Aplicar_Restriccion_Vehicular(char estado);
};

```

```

void Colectivo::Aplicar_Restriccion_Vehicular(char estado){
    if(estado == 'S' || estado == 's')
        Tiene_Restriccion = 'S';
    if(estado == 'N' || estado == 'n')
        Tiene_Restriccion = 'N';
}

class Particular : public Terrestre{
};

class Bus : public Colectivo{
private:
    int Valor_Pasaje_Estudiante;
public:
    int Obtener_Valor_Pasaje();
};

int Bus::Obtener_Valor_Pasaje(){
    return Valor_Pasaje_Estudiante;
}

class Automovil : public Colectivo{
};

class Maritimos : public Medios_Transporte{
protected:
    float Eslora;
};

class Privado : public Maritimos{
};

class Comercial : public Maritimos{
};

class Militar : public Maritimos{
private:
    int Numero_de_Canones;
};

class Artesanal : public Comercial{
};

class Industrial : public Comercial{
};

int main(void) {
    system("CLS");
    Particular MiMoto;
    Bus MiBus;
    Artesanal MiBote;
    MiMoto.Aumentar_Kilometraje_En(5);
    printf("Klms: %d", MiMoto.GetKilometraje());
    MiBus.CargarCombustible(20);
    MiBote.SetCapacidad(150);
    system("PAUSE");
}

```