

Nro. de orden: 24  
LU: 79/15  
Apellidos: Borghia  
Nombres: Agustín

1		2	3	4	5		TOTAL
a	b				a	b	
7	8	20	14	25	15		89

A

**Aclaraciones:** El parcial NO es a libro abierto. Cualquier decisión de interpretación que se tome debe ser aclarada y justificada. Para aprobar se requieren al menos 60 puntos. **Entregar cada ejercicio en hoja separada.**

**Importante:** Para la resolución del parcial NO es necesario ni está permitido el uso de **acum**. En caso de ser necesario, pueden asumir como definida la función *sum* ( $\Sigma$ ) sobre listas. Las funciones *min* o *max* sobre listas, en caso de requerirlas, deben ser definidas.

```
tipo Nombre=String;
tipo Año=Z;
tipo Categoría = MejorDirector, MejorPelícula, MejorActor, MejorActriz;
```

```
observador escenas (p: Película) : [Escena];
invariante ordenAlfabetico : ordenada(elenco(p));
invariante escenasConsecutivas : ...;
```

```
tipo Escena {
  observador numero (e: Escena) : Z;
  observador duracion (e: Escena) : Z;
  observador libreto (e: Escena) : String;
}
```

```
tipo Oscar {
  observador edicion (o: Oscar) : Año;
  observador premiosOtorgados (o: Oscar) : [Categoría];
  observador presentadores (o: Oscar) : [Nombre];
  observador nominadas (o: Oscar, c: Categoría) : [Película];
  requiere c ∈ premiosOtorgados(o);
  observador ganadora (o: Oscar, c: Categoría) : Película;
  requiere c ∈ premiosOtorgados(o);
  invariante nombresDistintos : (∀c ← premiosOtorgados(o))
  sinRepetidos([titulo(p)|p ← nominadas(o, c)]);
  invariante peliculasOk : ...;
```

```
tipo Película {
  observador titulo (p: Película) : Nombre;
  observador director (p: Película) : Nombre;
  observador estreno (p: Película) : Año;
  observador costo (p: Película) : Z;
  observador elenco (p: Película) : [Nombre];
```

```
aux sinRepetidos (xs: [T]) : Bool = (∀i, j ← [0..|xs|], i ≠ j) xsi ≠ xsj;
aux ordenada (xs: [T]) : Bool = (∀i ← [0..|xs| - 1]) xsi ≤ xsi+1;
```

**Ejercicio 1. [15 puntos]**

- Completar el invariante *escenasConsecutivas* del tipo *Película*, que indica que los números de las escenas de la película son consecutivos a partir del 1.
- Completar el invariante *peliculasOk*, que expresa que hay 3 películas nominadas por categoría, que la película ganadora de cada categoría había sido nominada para dicha categoría, y que el año de estreno de todas las películas corresponde a la edición en cuestión de los premios.

**Ejercicio 2. [20 puntos]** Especificar el problema *presentadorGanador* (os: [Oscar]) = result : Nombre, que devuelve el nombre de algún presentador en alguna de las ediciones de los Oscar de os que haya formado parte del elenco de alguna película ganadora en una edición anterior a la cual presenta.

**Ejercicio 3. [20 puntos]** Especificar el problema *baratasYganadoras* (o: Oscar) = result : [(Nombre, Z)], que teniendo en cuenta solamente las películas más baratas, devuelve las que más premios ganaron. Las tuplas devueltas contienen el nombre de la película y la duración de la misma (una película dura lo que suma la duración de sus escenas), ordenadas según la duración (de forma no decreciente).

**Ejercicio 4. [30 puntos]** Especificar el problema *renombrarGanadora* (o: Oscar, c: Categoría, t: Nombre), que modifica el Oscar cambiando el título de la película ganadora de la categoría c por t, en todos los lugares donde aparezca dicha película.

**Ejercicio 5. [15 puntos]** Dado el siguiente programa:

```
bool todosPares(int [] a, int n){
  bool result = true;
  if(a[0] % 2 != 0){
    result = false;
    a[0] = a[0] + 1;
  }
  return result;
}
```

- Dar la transformación de estados del programa presentado.
- Decidir si el programa es correcto para las siguientes especificaciones. Justificar.

```
problema todosPares1 (a: [Z], n: Z) = res : Bool {
  requiere |a| > 0;
  modifica a;
  asegura a == [parOEIQueSigue(x) | x ← pre(a)];
  asegura res == (∀x ← pre(a)) x mod 2 == 0;
}
```

```
problema todosPares2 (a: [Z], n: Z) = res : Bool {
  requiere |a| == 1;
  modifica a;
  asegura a == [parOEIQueSigue(x) | x ← pre(a)];
  asegura res == (∀x ← pre(a)) x mod 2 == 0;
}
```

```
aux parOEIQueSigue (x: Z) : Z = if x mod 2 == 0 then x else x + 1;
```