

LU:
Apellidos:
Nombres:
Orden:
Turno:

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.

Ejercicio 1. [50 puntos]

Dada la siguiente especificación:

```
problema factoriales (a:[Z], n : Z) {  
  requiere |a| == n ;  
  requiere n ≥ 1 ;  
  requiere a == [1..n] ;  
  modifica a ;  
  asegura |a| == n ;  
  asegura (∀i ← [0..n]) ai == fact(i + 1) ;  
  aux fact (n:Z) : Z = ∏[i | i ← [1..n]] ;  
}
```

Se propone el siguiente código:

```
void factoriales(int a[], const int n) {  
  int i = 1 ;  
  while (i < n) {  
    a[i]=a[i-1]*a[i];  
    i++;  
  }  
}
```

- Escribir la transformación de estados del código dado.
- Considerar el siguiente invariante para el ciclo: $I : 1 \leq i \leq n \wedge a == [fact(j + 1) | j \leftarrow [0..i]] + +[a@Pc_j | j \leftarrow [i..n]]$. Demostrar que el cuerpo del ciclo preserva el invariante.
- Mostrar que $Pc \Rightarrow I$

Ejercicio 2. [20 puntos]

Escribir la especificación del ciclo (Pc, I, Qc, función variante y su cota) que permita la demostración de correctitud de la implementación dada para el problema multiplicarSimetrico. Suponer que existe la siguiente precondition: requiere: |a| == n. No se pide realizar la demostración.

```
void multiplicarSimetrico(int a[], const int n) {  
  int i = 0 ;  
  while (i < n/2) {  
    a[i] = a[i] * a[n-1-i];  
    a[n-1-i] = a[i];  
    i++;  
  }  
}
```

Ejercicio 3. [30 puntos]

Dada la siguiente especificación para el problema modificacionLocaArreglo:

```
problema modificacionLocaArreglo (a: [Z], n: Z) {  
  requiere |a| == n ;  
  requiere n mod 2 == 1 ;  
  modifica a ;  
  asegura a == [modificacionLoca(pre(a)i, i) | i ← [0..n]] ;  
  aux modificacionLoca (x, i: Z) : Z = if (i mod 2 == 0) then -1 * x else 0 ;  
}
```

- Dar una implementación del problema modificacionLocaArreglo que utilice un único ciclo que respete el siguiente invariante:
 $I : 0 \leq i < n \wedge i \text{ mod } 2 == 0 \wedge$
 $(\forall j \leftarrow [0..i]) a_j == pre(a)_j \wedge (\forall j \leftarrow (i..n)) a_j == modificacionLoca(pre(a)_j, j)$
- Escribir Pc, Qc, B, una función variante y su cota para el código dado, de forma que, usando el invariante planteado, se pueda demostrar la correctitud del ciclo. (No se pide realizar la demostración.)