1.– A partir de la siguiente especificación de los números naturales

```
modulo NATURALES
  generos
    Nat
  operaciones
    cero: -> Nat
    suc:  Nat -> Nat
    mas:  Nat Nat -> Nat
  variables
    x,y:Nat
  ecuaciones
    mas(x,cero)=x
    mas(x,suc(y))=suc(mas(x,y))
finmodulo
```

se pide:

1. Especificar formalmente la operación producto de naturales.
2. Demostrar formalmente (utilizando inducción estructural) a partir de tal especificación las propiedades conmutativa y asociativa del producto de naturales.
3. Especificar la función `es_par:Nat -> Bool` que comprueba si un número dado es par.
4. Especificar la función `sig_par:Nat -> Nat` que, dado un número natural, calcula el menor número natural par que sea estrictamente mayor que él.

2.– Realizar una especificación algebraica de los polinomios $\mathbb{Z}[X]$ de una variable con coeficientes enteros, es decir

$$p(x) = a_0 + a_1x + a_2x^2 + \cdots + a_nx^n, \quad 0 \leq n, \quad a_i \in \mathbb{Z} \quad \forall i \in \{0..n\}$$

con las operaciones:

- `cero`: construye el polinomio nulo $p(x) = 0$.
- `añade`: que, dado un polinomio $p(x)$ y un par (a, i) con $a \in \mathbb{Z}, i \in \mathbb{N}$, devuelve $p(x) + ax^i$.
- `evalua`: que, dado un polinomio $p(x)$ y un entero $b \in \mathbb{Z}$ devuelve $p(b)$, es decir, el resultado de evaluar el polinomio p en el valor b.

3.– Apoyándose en el módulo especificado en el problema anterior, especificar formalmente otro módulo de librería sobre los polinomios $\mathbb{Z}[X]$ que aporte las siguientes funciones:

- `coeficiente`: dado un polinomio $p(x)$ y un natural i, devuelve el coeficiente de $p(x)$ de grado i (es decir, a_i).
- `suma`: dados dos polinomios $p(x)$ y $q(x)$, devuelve el polinomio suma $s(x) = p(x)+q(x)$.
- `mult`: dados dos polinomios $p(x)$ y $q(x)$, devuelve el polinomio producto $r(x) = p(x)q(x)$.

4.– Realizar una especificación formal de los números enteros con las operaciones `cero`, `suc` (sucesor), `pred` (predecesor) y `suma` (suma de dos enteros).

5.– Apoyándose en la especificación del ejercicio anterior:

1. Demostrar la conmutatividad de la suma de enteros.
2. Especificar la función `producto` de enteros y demostrar si es conmutativa o no.
3. Especificar la función `resta` y demostrar si es conmutativa o no.
4. Especificar la función `menoroigual:Ent Ent -> Bool` y, apoyándose en ella, especificar las funciones `mayor` e `igual`.
5. Especificar la función `neg:Ent -> Ent` que, dado un entero, le cambia su signo (es decir, `neg(8)=-8`, `neg(-3)=3`).

6.– Se desea especificar algebraicamente un diccionario de palabras a partir de la siguiente signatura:

```
modulo DICCIONARIO
  importa Palabra de PALABRAS, Bool de BOOL
  generos
    Dic
  operaciones
    crea: -> Dic
    ins:  Dic Palabra -> Dic    (* inserta una palabra *)
    borr: Dic Palabra -> Dic    (* borra una palabra *)
    esta: Dic Palabra -> Bool   (* comprueba si una palabra esta *)
finmodulo                       (* o no en el diccionario *)
```

Se pide realizar una posible especificación de tal tipo de datos, en donde también han de especificarse todos aquellos tipos abstractos de los que haga uso.

7.– Especificar una solución del problema de Hanoi para tres varillas con N aros basándose en la siguiente signatura:

```
modulo HANOI
  importa Nat de NATURALES
  generos
    Varilla, Hanoi
  operaciones
    N:         -> Nat
    a,b,c:     -> Varilla
    inicio:    Nat -> Hanoi
    pasar:     Varilla Varilla Hanoi -> Hanoi
    sol_parc: Varilla Varilla Varilla Nat Hanoi -> Hanoi
    resolver: Nat Hanoi -> Hanoi
finmodulo
```

8.– Especificar el tipo abstracto de datos "`Z8`" que representa al grupo abeliano $(\boldsymbol{Z}_8, +)$ de 8 valores {0,`suc(0)`,...,`suc`^{7}`(0)`} con generadores `0:-> Z8` y `suc:Z8 -> Z8` y la operación `suma:Z8 Z8 -> Z8`. Especificar también las operaciones `pred` y `producto` sobre `Z8`.