Segundo Parcial – Tema 1

- **NO OLVIDAR PONER EL NOMBRE Y APELLIDO EN CADA HOJA**
- **NUMERAR LAS HOJAS CON LA FORMA <hoja número>/<cantidad de hojas>**
- **DEJAR MEDIA CARILLA EN BLANCO EN LA PRIMERA HOJA.**
- Aconsejamos leer el examen completo antes de empezar a resolver los ejercicios, ya que ayuda a comprender mejor el dominio. Se puede consultar cualquier material (en papel) que haya sido escrito antes de comenzar el examen y usar sin definir todas las funciones y procedimientos vistos durante la cursada
- Pensar bien la estrategia a seguir,consultar lo que no se entienda y recordar que el parcial es una instancia más de aprendizaje donde algún docente va a dar una devolución personalizada de todo lo que se escriba, es necesario para aprovechar esta instancia como algo más que solamente un momento para sacar una nota.

GobstoMon

Los GobstoMon son una criaturas que viven en un mundo paralelo. Los seres humanos, a la temprana edad de 12 años se hacen llamar “Entrenadores”, y capturan a esas pobres criaturas en unas pequeñas jaulas esféricas donde los dejan sin agua ni comida por días, para luego soltarlos a batallar contra los GobstoMons de otros entrenadores cual gallos de rapiña, y declararse como el maestro GobstoMon supremo. Muy turbio, lo sabemos, pero así es este universo paralelo.

Así, nos interesará modelar en Gobstones esté oscuro pero interesante universo. Para ello contaremos con las siguientes definiciones de tipos:

```
type TipoDeGobstoMon is variant {
    case Fuego  {}
    case Agua   {}
    case Planta {}
}
```

```
type GobstoMon is record {
    field nombre    // String
    field tipo      // TipoDeGobstoMon
    field poder     // Número > 0
}
```

```
type Entrenador is record {
    field nombre      // String
    field gobstomons  // Lista de GobstoMon
}
```

Deseamos entonces realizar una serie de funciones y procedimientos sobre estos datos.

Ejercicio 1)

Cuando los GobstoMons pelean entre si, uno puede tener ventaja sobre otro, y esto depende de los tipos de ambos GobstoMons. Tenemos la siguiente tabla de ventajas:

Tipo:	Tiene ventaja contra:
Agua	Fuego
Fuego	Planta
Planta	Agua

En ese sentido, queremos realizar las siguientes tres funciones:

a. **aguaTieneVentajaContra(otroTipo)** que describe Verdadero si "otroTipo" es un tipo contra el cual el tipo Agua tiene ventaja. Falso en caso contrario.
b. **fuegoTieneVentajaContra(otroTipo)** que describe Verdadero si "otroTipo" es un tipo contra el cual el tipo Fuego tiene ventaja. Falso en caso contrario.
c. **plantaTieneVentajaContra(otroTipo)** que describe Verdadero si "otroTipo" es un tipo contra el cual el tipo Planta tiene ventaja. Falso en caso contrario.

Ejercicio 2)

Se desea ahora una función un poco más genérica que la anterior, **tipoTieneVentaja_Contra_(unTipo, otroTipo)** que dados dos tipos, indica si el primero tiene ventaja sobre el segundo. Piense en utilizar las funciones del ejercicio anterior.

Ejercicio 3)

También quisieramos saber si, dados dos GobstoMons, el primero tiene ventaja sobre el segundo. Para ello, se desea la función tiene **tieneVentaja_Contra_(unGobstoMon, otroGobstoMon)**

Ejercicio 4)

Queremos ahora crear algunos GobstoMons para probar nuestro trabajo. Así, proponemos que se creen tres funciones, que retornan un GobstoMon:

1. **garmander()** Que retorna un GobstoMon de tipo Fuego, con poder 5 y con nombre "Garmander".
2. **golbasaur()** Que retorna un GobstoMon de tipo Planta, con poder 4 y con nombre "Golbasaur".
3. **gobartle()** Que retorna un GobstoMon de tipo Agua, con poder 6 y con nombre "Gobartle".

Ejercicio 5)

Cuando dos GobstoMons pelean ganará aquel que tenga más poder. Sin embargo, si un GoibstoMon tiene ventaja sobre otro, se le deben adicionar 2 puntos de poder adicionales. Por ejemplo, si pelean Golbasaur contra Gobartle, Golbasaur tendrá ventaja contra Gobartle, y su poder total será 6. Así, la pelea resultará en empate.

Considerando el siguiente tipo de datos:

```
type ResultadoDeBatalla is variant {
    case GanaPrimero  {}
    case GanaSegundo  {}
    case Empatan      {}
}
```

Se desea realizar la función **quienGanaBatallaEntre_Y_(primerGobstoMon, segundoGobstoMon)** que retorna un ResultadoDeBatalla determinando si el primer GobstoMon le ganó al segundo, si fue el segundo el que ganó o si empataron.

Ejercicio 6)

Se desea realizar una función **nombresDeLosGobstoMonDe_(unEntrenador)** que, dado un entrenador, describa una lista de los nombres de los GobstoMon de dicho entrenador (solo los nombres, o sea, una lista de Strings).

Ejercicio 7)

Se desea realizar una función **soloLosDeTipo_De_(unTipo, unEntrenador)** que, dado un entrenador, describa una lista de los GobstoMon de dicho entrenador que tienen el tipo dado.

Ejercicio 8)

Se desea saber si un entrenador es "malo". Los GobstoMon de un entrenador malo son todos del mismo tipo.

Se desea entonces la función **esEntrenadorMalo(unEntrenador)** que describe Verdadero si cada GobstoMon del entrenador dado tiene el mismo tipo. Puede asumir que el entrenador dado tiene al menos un GobstoMon.