

Apellidos:**Nombre:****Grupo 1C**

Ejercicio 1 Define la función

```
format :: String -> String
```

que formatea la cadena `xs` eliminando los espacios iniciales y finales, así como espacios superfluos. Por ejemplo:

```
format " El barco ebrio " ==> "El barco ebrio"
```

Se define el tipo de dato

```
type Var = Char
```

```
data Form = V Var | No Form | O Form Form | Y Form Form
```

```
deriving Show
```

para representar fórmulas de la lógica proposicional.

Ejercicio 2 Define la función

```
sust :: (Var, Var) -> Form -> Form
```

tal que `(sust (x,y) g)` devuelve la fórmula resultante de sustituir cada ocurrencia de la variable `x` en `g` por la variable `y`. Por ejemplo:

```
sust ('p','q') (O (No (V 'p')) (O (V 'r') (V 'p')))) ==>
O (No (V 'q')) (O (V 'r') (V 'q'))
```

Un literal es una fórmula atómica o la negación de una fórmula atómica. Una fórmula está en forma normal disyuntiva (FND) si es una disyunción de conjunciones de literales.

Ejercicio 3 Define el predicado

```
esFND :: Form -> Bool
```

que determine si una fórmula está en forma normal disyuntiva.

```
esFND (Y (No (V 'p')) (O (V 'r') (V 'p')))) ==> False
```

```
esFND (O (No (V 'p')) (Y (V 'r') (V 'p')))) ==> True
```