

Laboratorul 10 - Functor

```
{-  
class Functor f where  
fmap :: ( a -> b ) -> f a -> f b  
-}
```

Scrieți instanțe ale clasei `Functor` pentru tipurile de date descrise mai jos.

```
newtype Identity a = Identity a
```

```
data Pair a = Pair a a
```

```
data Constant a b = Constant b
```

```
data Two a b = Two a b
```

```
data Three a b c = Three a b c
```

```
data Three' a b = Three' a b b
```

```
data Four a b c d = Four a b c d
```

```
data Four'' a b = Four'' a a a b
```

```
data Quant a b = Finance | Desk a | Bloor b
```

S-ar putea să fie nevoie să adăugați unele constrângeri la definirea instanțelor

```
data LiftItOut f a = LiftItOut (f a)
```

```
data Parappa f g a = DaWrappa (f a) (g a)
```

```
data IgnoreOne f g a b = IgnoringSomething (f a) (g b)
```

```
data Notorious g o a t = Notorious (g o) (g a) (g t)
```

```
data GoatLord a = NoGoat | OneGoat a | MoreGoats (GoatLord a) (GoatLord a) (GoatLord a)
```

```
data TalkToMe a = Halt | Print String a | Read (String -> a)
```