

Swift switch and enum

Problem II

列舉(enum)型態Layer包含五種不同的互斥狀況，請在第三行補足狀況欄位。
請在第五行將變數layer的型態宣告為Layer，並根據列印結果設定初始狀況欄位的初始值。
請在第十行完成switch指令的狀況欄位指定。

```
1 import UIKit
2 enum Layer{
3     case input,         , pooling
4     case relu, maxsoft|
5 }
6 var layer =                 
7 switch layer{
8 case .input: print("an image")
9 case .conv: print("filter convolution")
10 case           : print("unitary activations")
11 case .pooling: print("down sampling")
12 case .relu: print("nonlinear")
13 }
```

conv

"filter convo...

filter convolution

本題的列舉(enum)型態Layer有五個狀況欄位，還有一個稱為note的方法。本題在方法note中使用switch指令，根據狀況欄位的內容，列印不同的結果，請完成第五行的switch指令

請在第七行中完成layer變數的初始設定

```
1  enum Layer{
2      case input,           , pooling
3      case relu, softmax
4      func note(){
5                            {
6              case .input: print("an image")
7              case .conv: print("filter convolution")
8              case .softmax: print("unitary activations")
9              case .pooling: print("down sampling")
10             case .relu: print("nonlinear")
11             }
12         }
13     }
14     var layer = Layer.                  
15     layer.note()
```

"unitary acti...

softmax

softmax

unitary activations

請將本題的列舉(enum)Label宣告為整數型態，兩個狀況欄位benign與malignant的原值(rawValue)，分別為整數1與2
請參考第六行常數label的原值列印結果，在第五行中宣告常數label為列舉(enum)Label，並設定列舉狀況

```
1 enum ██████████ {  
2     case benign = 1  
3     case malignant = 2  
4 }  
5 let label = Label.██████████  
6 print(label.rawValue)
```

benign
"1\n"



1

本題的列舉(enum)Course有六個狀況欄位，原值為字串型態，請第一行與第六行完成Course的宣告指令。

請根據第十行的原值列印結果，在第八行將常數myCourse的型態，宣告為列舉(enum)Course，並設定其初始狀況

```
1 enum Course: ██████████ {
2     case la = "linear algebra"
3     case ca = "calculus"
4     case dm = "discrete mathematics"
5     case nm = "numerical methods"
6     case nn = ██████████
7     case ra = "real analysis"
8 }
9 let myCourse = ██████████
10 let fullCourse = myCourse.rawValue
```

```
nn
"neural net..."
```



neural networks

本題的列舉(enum)Course有兩個狀況欄位，分別具備不同的原值型態，其中狀況欄位id的原值為整數型態，abbrev的原值為字串型態。

```
1 import UIKit
2 enum Course{
3     case id( )
4     case abbrev( )
5 }
```

本題的列舉(enum)Course有兩個狀況欄位，分別具備不同的原值型態，其中狀況欄位id的原值為整數型態，abbrev的原值為字串型態。

請根據第八行與第九行的原值列印結果，在第六行將常數myCourse的型態，宣告為列舉(enum)Course，同時設定其初始狀況與原值。

請參考第九行的寫法，在第八行將常數id設定為列舉變數myCourse的原值

```
1 import UIKit
2 enum Course{
3     case id( )
4     case abbrev( )
5 }
6 let myCourse = 
7 switch myCourse{
8 case .id( ): print(id)
9 case .abbrev(let abbrev): print(abbrev)
10 }
```



id(...)

"10..."

本題的列舉(enum)Course有兩個狀況欄位，分別具備不同的原值型態，其中狀況欄位id的原值為整數型態，abbrev的原值為字串型態。

請根據第八行與第九行的原值列印結果，在第六行將常數myCourse的型態，宣告為列舉(enum)Course，同時設定其初始狀況與原值。

請參考第八行的寫法，在第九行將常數abbrev設定為列舉變數myCourse的原值

```
1 import UIKit
2 enum Course{
3     case id( )
4     case abbrev( )
5 }
6 let myCourse = 
7 switch myCourse{
8 case .id(let id): print(id)
9 case .abbrev( ): print(abbrev)
10 }
```



ab...

"la\n"

本題的列舉(enum)Day有七個狀況欄位，原值型態為整數，請將狀況欄位Mon的原值設定為1。其它狀況欄位的原值將自動設定。

請根據第十二行的原值列印結果，在第十一行將變數day的型態，宣告為列舉(enum)Day，同時設定其初始狀況。

請參考第八行的寫法，在第九行將常數abbv設定為列舉變數myCourse的原值

```
1 import UIKit
2 enum Day: Int{
3     case Mon =           
4     case Tue
5     case Wed
6     case Thu
7     case Fri
8     case Sat
9     case Sun
10 }
11 var day = Day.Sun
12 print(day.          )
```

Sun

"7\n"

