

Swift struct

宣告結構(struct)Course，結構名稱的第一個字母通常使用大寫，其餘小寫。本題的結構有兩個欄位，一個是整數型態的變數 id，另一個是字串型態的變數title，

```
1 import UIKit
2 struct Course{
3                     
4     var title: String
5 }
```

型態結構(struct)Course宣告後，可以用來定義變數。請宣告變數course，並將其型態宣告為結構(struct)Course，同時將該結構的欄位變數id設定為10，並將欄位變數title設定為字串"ca"

```
1 import UIKit
2 struct Course{
3     [redacted]
4     var title: String
5 }
6 var [redacted]
7 print("course id:", course.id, "title: ", course.title)
```

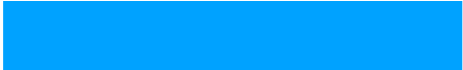
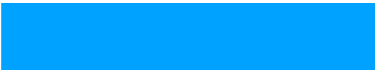

本題作答部分

Course
"course..."

course id: 10 title: ca

本題宣告列舉(enum)型態Activation，以及結構(struct)型態Neuron。其中列舉型態Activation有四種互斥狀況，分別為linear，threshold，tanh，以及sigmoid，代表神經元運算的四種活化函數。

結構型態Neuron有兩個欄位變數，欄位變數weight的型態宣告為double實數陣列，欄位變數activation的型態為列舉(enum)Activation

```
1 import UIKit
2 enum Activation{
3     case linear
4     case threshold
5     
6     case sigmoid
7 }
8 struct Neuron{
9     var 
10    var 
11 }
```

使用init進行結構(struct)Neuron的欄位變數初始化，本題的init方法有兩個輸入參數，參數名稱分為weight及activation，其中weight為Double實數陣列，而activation的型態為列舉(enum) Activation。

在設定時，分別將self.weight及self.activation所代表的兩個欄位變數，指定為輸入參數weight及activation的內容

```
1 import UIKit
2 enum Activation{
3     case linear
4     case threshold
5     case tanh
6     case sigmoid
7 }
8 struct Neuron{
9     var           
10    var activation: Activation
11    init(          ){
12        self.weight = weight
13                   = activation
14    }
15 }
```

本題作答部分

當結構Neuron的欄位變數及初始方法完成後，即可在第11行將常數neuron的型態指定為結構(struct)Neuron。請根據print指令的輸出結果，在第16行設定欄位變數weight及activation的初始內容

```
1 import UIKit
2 enum Activation{
3     case linear
4     case threshold
5     [redacted]
6     case sigmoid
7 }
8 struct Neuron{
9     var weight: [Double]
10    var activation: Activation
11    init([redacted]){
12        self.weight = weight
13        [redacted] = activation
14    }
15 }
16 let [redacted] = Neuron([redacted])
17 print(neuron.weight)
18 print(neuron.activation)
```



本題作答部分

Neuron
"[1.0,...
"tanh\n"

☐ ☐

[1.0, 2.0]
tanh

請在第7-8行init方法的主體命令中，設定結構(struct)Line的兩個變數欄位的初始內容。

```
1 import UIKit
2
3 struct Line{
4     var slope: Double
5     var intercept: Double
6     init(slope: Double, intercept: Double){
7         
8         
9     }
10 }
```

請根據第12行命令及輸出視窗的內容，將常數line之型態宣告為結構(struct)Line，並設定欄位變數之初始內容。

```
1 import UIKit
2
3 struct Line{
4     var slope: Double
5     var intercept: Double
6     init(slope: Double, intercept: Double){
7         _____
8         _____
9     }
10 }
11 let _____
12 print(line.slope, line.intercept)
14
```

本題作答部分

Line
"1.0 1...."

第8行已經宣告變數line的型態為結構(struct)Line，並設定相關欄位變數之初始內容。請在第9行將欄位變數intercept的內容設定為1

```
1 import UIKit
2
3 struct Line{
4     var slope: Double
5     var intercept: Double
6 }
7
8 var line = Line(slope: 1, intercept: 0)
9
10 print(line.slope, line.intercept)
12
```

Line

Line

"1.0 1...."

結構(struct)Line有三個欄位變數，方法init僅設定兩個欄位變數的初始內容，並未設定passOrigin的初始內容，請在第6行宣告passOrigin，並將其初始內容設定為false。請完成print指令，印出三個欄位變數之內容

```
1 import UIKit
2
3 struct Line{
4     var slope: Double
5     var intercept: Double
6     var passOrigin: Boolean
7     init(slope: Double, intercept: Double){
8         self.slope = slope
9         self.intercept = intercept
10    }
11 }
12
13 let line = Line(slope: 2, intercept: 1)
14 print(line)
```

Line
"2.0 1....

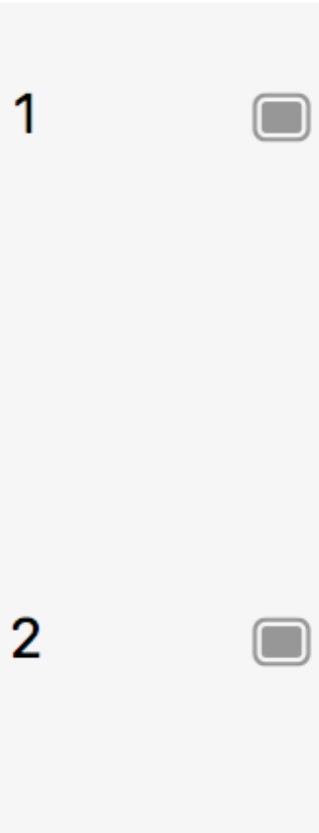


2.0 1.0 false

變數idGen初始值為1，結構(struct)Student有兩個欄位變數，分別為整數常數studentId及字串變數name，其中studentId為常數，內容不可改變。

結構(struct)Student的init方法使用idGen的內容設定整數常數studentId，設定後idGen的內容自動加1。

```
1 import UIKit
2 var idGen = 1
3 struct Student{
4     let studentId: Int
5     var name: String
6     init(name: String){
7         self.name = name
8         [REDACTED]
9         idGen += 1
10    }
11 }
```



請根據第13行命令及輸出視窗的內容，將變數student1之型態宣告為結構(struct)Student，並設定欄位變數之初始內容。

```
1 import UIKit
2 var idGen = 1
3 struct Student{
4     let studentId: Int
5     var name: String
6     init(name: String){
7         self.name = name
8         _____
9         idGen += 1
10    }
11 }
12 var _____
13 print(student1.studentId, student1.name)
```

本題作答部分

1

2

Student

"1 John..."

☐

1 John Lin

請在第4行將欄位變數major設定為optional字串變數

```
1 import UIKit
2 struct Student{
3     var name: String
4     var           
5     init(name: String){
6         self.name = name
7     }
8 }
```

請在第10行將常數student1Major的內容指定為student1欄位變數major之內容。本題指定失敗，印出結果為"unspecified"

```
1 import UIKit
2 struct Student{
3     var name: String
4     var                     
5     init(name: String){
6         self.name = name
7     }
8 }
9 var student1 = Student(name: "John Lin")
10 if                                              {
11     print(student1Major)
12 }
13 else{
14     print("unspecified")
15 }
```



Student

"unspec...



unspecified

本題使用兩個init方法，第5-7行，僅設定欄位變數name的初始內容，第8-11行指定兩個欄位變數之內容，請完成第二個init方法

```
1 import UIKit
2 struct Student{
3     var name: String
4     var major: String?
5     init(name: String){
6         self.name = name
7     }
8     init(name: String, major: String){
9         
10        
11    }
12 }
```

當結構Student的欄位變數及初始方法完成後，即可在第13行將變數student1的型態指定為結構(struct)Student。請根據print指令的輸出結果，在第13行設定欄位變數name及major的初始內容

```
1 import UIKit
2 struct Student{
3     var name: String
4     var major: String?
5     init(name: String){
6         self.name = name
7     }
8     init(name: String, major: String){
9                         
10                        
11    }
12 }
13 var                 
14 if let student1Major = student1.major{
15     print(student1.name, student1Major)
16 }
```



Student

"John Li...



請在Xcode實作本題，並簡單說明程式內容

```
1 import UIKit
2
3 struct Line{
4     var slope: Double
5     var intercept: Double
6     mutating func shiftTo(x: Double, y: Double){
7         self.intercept = y - self.slope * x
8     }
9 }
10
11 var line = Line(slope: 1, intercept: 0)
12 line.intercept = 1
13 line.shiftTo(x: 2, y: 1 )
14 print("slope:", line.slope, "intercept: ", line.intercept)
```

Line

Line

Line

Line

"slope:..."

slope: 1.0 intercept: -1.0

`line.shiftTo(x: 2, y: 1)`