

Swift optional variables and random generator

本題字串變數char的內容均為數字字元，第5行Int函數轉換成功字串char，變數value的內容為整數。變數value在第5行的指定內容為Int轉換結果，有未定義的可能，須設定為optional(可選擇的)變數，才能通過編譯。請在第6行設定條件指令的檢查條件，並在第9行，列印變數value的展開(unwrap)內容，展開時，請記得在變數value後加驚嘆號“!”。

```
1 import UIKit
2
3 var char: String = "123"
4 var value: Int?
5 value = Int(char)
6 if [REDACTED] {
7     print("variable value is undefined")
8 } else{
9     print([REDACTED])
10 }
```

```
"12...
nil
123
```

```
"12...
```

請在第2行，將常數cardNumber宣告為整數Int，並使用Int.random的方法隨機設定初始值，請使用函數參數in指定1到52為該方法產生隨機整數之範圍

```
1 import UIKit
2 let cardNumber =
```

本題宣告變數 `cardNumbers` 為包含 1 到 52 的整數陣列，請使用該陣列的 `shuffled()` 方法將陣列內容隨機排列

```
1 import UIKit
2
3 let cardNumbers = 1...52
4 var card =
```

```
41
{lowe...
[42,...
```

常數geneCharSet為字串陣列，內容包含基因序列的4個字元。請將geneChar宣告為optional字串變數，並以字串陣列geneCharSet的randomElement()方法隨機產生字元，設定geneChar的初始值

```
1 import UIKit
2
3 let geneCharSet = ["A", "T", "C", "G"]
4 var geneChar: 
```

```
["A",...
"G"]
```

承接上一題，geneChar為optional字串變數，並以字串陣列geneCharSet的randomElement()方法隨機設定初始值，在geneChar變數內容展開(unwrap)時，有可能未被定義。本題以條件指令，判斷geneChar是否不等於nil，當條件成立後，才會執行print指令，印出變數geneChar的展開內容，展開geneChar時，請記得在該變數後加驚嘆號“!”。

```
1 import UIKit
2
3 let geneCharSet = ["A", "T", "C", "G"]
4 var geneChar:
5 if
6     print(
7 }|
```

請填這兩行

```
["A",...
"T"
"T\n"
```

本題隨機產生包含10個基因字元的字串變數geneString，並列印。
第6行中for的索引變數，並未在迴圈中使用，以符號“_”替代，請以1...10為迴圈索引範圍完成for指令設計，在第7行使用字串陣列geneCharSet的randomElement()方法隨機產生字元，在第8行，以條件指令判斷geneChar是否不等於nil。

```
1 import UIKit
2
3 let geneCharSet = ["A", "T", "C", "G"]
4 var geneChar: String?
5 var geneString: String = ""
6 for _ in 1...10 {
7     geneChar = geneCharSet.randomElement()
8     if geneChar != nil {
9         geneString += geneChar!
10    }
11 }
12 print("geneString: " + geneString)
```

```
["A...
nil
""
(10...
(10...
"...
```

陣列charSet有兩個字串，其中一個包含非數字字元。第4行使用randomElement()的方法，隨機選定一個字串，代入字串變數char中。請在第5行將num宣告為optional整數，並使用Int函數將展開後的char內容轉換為整數，請記得在char之後緊接符號"! "。如果隨機產生的char包含非字元字串，則num未定義。請在第6行使用條件指令，檢查num是否等於nil，當條件不成立時才執行num!的列印。

```
1 import UIKit
2
3 let charSet = ["1", "T"]
4 var char: String? = charSet.randomElement()
5 var num: Int
6 if {
7     print("num does not exist")
8 } else {
9     print(num!)
10 }
```

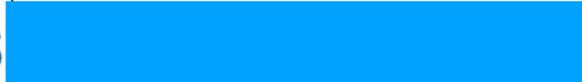
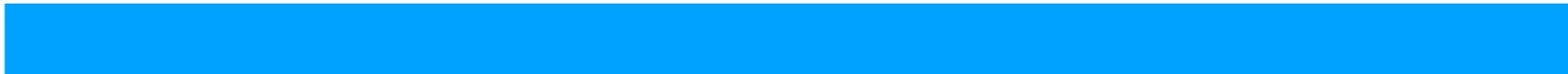
```
["1...
"1"
1
"1\n"
```

請在第5行將num宣告為optional整數，並使用Int函數將展開後的char內容轉換為整數，請記得在char之後緊接符號"! "。如果隨機產生的char包含非字元字串，則num未定義。請在第6行使用條件指令，使用let宣告常數c，並將num的內容指定到c，如果num有內容，指定成功並執行第7行，否則執行第9行。

```
1 import UIKit
2
3 let charSet = ["1", "T"]
4 var char: String? = charSet.randomElement()
5 var num: Int
6 if {
7     print(c)
8 } else {
9     print("char does not exist")
10 }
```

```
["1...
"1"
1
"1\n"
```

pointSet為整數陣列常數，代表骰子六個面的點數。請在第4行將**tenPoints**宣告為整數陣列，並初始化為空陣列；並在第6行條件指令中，以**let**宣告常數**point**，使用陣列**pointSet**的**randomElement()**方法隨機產生一整數，指定為常數**point**的內容，指定成功後就能執行第7行。請在第7行將**point**的內容附加在陣列**tenPoints**之後。

```
1 import UIKit
2
3 let pointSet = 1...6
4 var tenPoints 
5 for _ in 1...10{
6     if  {
7         
8     }
9 }
10 print(tenPoints)
```

{10...
[]

(10...
"[3,...

承接上題，tenPoints代表隨機產生的骰子點數。請在第11行，以num為列舉變數，使用for指令列舉整數陣列tenPoints的內容；在第12行以條件指令判斷num是否等於1，並在第13行更新countOne的計數個數

```
1 import UIKit
2
3 let pointSet = 1...6
4 var tenPoints [Int]
5 for _ in 1...10{
6     if [Int] {
7         [Int]
8     }
9 }
10 var countOne = 0
11 for num in tenPoints {
12     if num == 1 {
13         countOne += 1
14     }
15 }
16 print(countOne)
```

本題請填

```
{lo...
[]
(10...
0
1
"1\n"
```