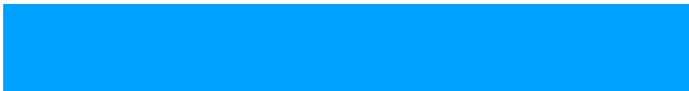


Swift loop

for 迴圈使用變數number列舉陣列中的儲存內容，依序代入迴圈，進行迭代運算。本題陣列變數index存儲5個整數，請在for迴圈指令中，使用變數number，列舉陣列index內容，進行迴圈迭代運算

```
1 import UIKit
2
3 var index = 1...5
4 for                      {
5     print(number*number)
6 }
```

陣列變數index儲存5個整數， for迴圈指令使用變數number列舉index陣列內容，如何設計迴圈的主體指令，進行index陣列的數字加總呢？而且以變數sum儲存加總結果。

```
1 import UIKit
2
3 var index = 1...5
4 var sum = 0
5 for number in index{
6     
7 }
8 print(sum)|
```

在for迴圈指令中，使用numStart與numEnd設計閉區間，指定變數number的範圍，印出4個數字，如輸出視窗所示

```
1 import UIKit
2
3 let numEnd = 5
4 let numStart = 2
5 {
6     print(number)
7 }
```



```
5 (
2 (
(4 ti... (
```



```
2
3
4
5
```

在for迴圈指令中，使用numStart與numEnd設計半閉區間，指定變數number的範圍，印出3個數字，如輸出視窗所示

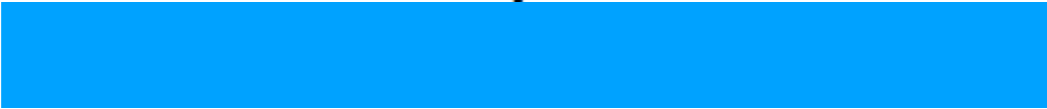
```
1 import UIKit
2
3 let numEnd = 5
4 let numStart = 2
5 for number in numStart..
```

5
2
(3 ti...)



2
3
4

陣列變數x儲存5個數字，for迴圈指令以變數number列舉陣列x中的所有數字，依序代入迴圈。如何設計迴圈主體運算，將陣列x中的數字依序加總至變數su?

```
1 import UIKit
2
3 var x = [1, 2, 8, 4, 5]
4 var su = 0
5 for number in x{
6     
7 }
8 print(su)
```

```
[1, 2, 8, 4, 5]
0
(5 times)
"20\n"
```

本題透過for迴圈列舉字串陣列deepNN中的字串，請在for迴圈指令中，以變數layer依序列舉deepNN各個字串。

```
1 import UIKit
2
3 var deepNN = ["input", "hidden1", "hidden2", "output"]
4 for layer in deepNN {
5     print(layer)
6 }
```

第三行透過 `Array` 指令設定陣列變數 `tenDoubles`，以 `repeating` 設置實數 `0.0`，重複次數為 `10`。請以類似指令設定陣列變數 `anotherTenVariables` 的內容，為包含 `10` 個實數 `1.5` 的陣列，成功設定後，`for` 迴圈將進一步加總陣列中的所有數字，總和為 `15`，如第九行右邊的變數內容所示

```
1 import UIKit
2
3 var tenDoubles = Array(repeating: 0.0, count: 10)
4 var [REDACTED]
5 var sum = 0.0
6 for number in anotherTenVariables{
7     sum += number
8 }
9 print(sum)
```

```
[0, 0, 0, ...
[1.5, ...
0
(10 times)
"15.0\n"
```


使用附加指令可以將兩個陣列，合併為一個陣列。
`tenDoubles`和`anotherTenVariables`分別儲存10個數字，請
使用附加指令，合併兩個數字陣列，並儲存在陣列變數
`twentyVariables`中。經迴圈運算加總，得到總和35

```
1 import UIKit
2
3 var tenDoubles = Array(repeating: 2.0, count: 10)
4 var anotherTenVariables = Array(repeating: 1.5, count: 10)
5 var twentyVariables = Array(repeating: 2.0, count: 10)
6 var sum = 0.0
7 for number in twentyVariables{
8     sum += number
9 }
10 print(sum)
```

[2, 2, 2,...
[1.5,...
[2, 2, 2,...
0
(20 times)
"35.0\n"


tuple(元組)是以小括號將多重變數組合所形成的資料型態，例如，在**for**迴圈中，可以使用**(index, layer)**形成雙變數組合，列舉**deepNN.enumerated()**的內容。列舉結果，**index**儲存列舉的字串編號，而**layer**存儲列舉字串，依序代入迴圈中。請以雙變數組合**(index, layer)**，列舉**deepNN**的儲存內容。透過迴圈主體命令，將**index**和**layer**的列舉內容依序代入列印指令，印出結果，如輸出視窗所示。

```
1 import UIKit
2
3 var deepNN = ["input", "conv", "ReLu", "pooling", "conv", "softMax"]
4 for                  {
5     print("        layer "+String(index)+" is "+layer)
6 }
```



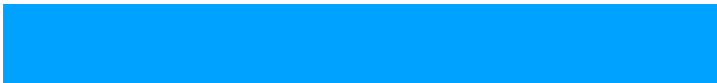
```
layer 0 is input
layer 1 is conv
layer 2 is ReLu
layer 3 is pooling
layer 4 is conv
layer 5 is softMax
```

maxLoopNum代表最大迴圈數，**loop**代表實際執行的迴圈數，未進入迴圈前，初始值為0。while迴圈指令需設計進入條件，請完成while迴圈指令設計，當迴圈執行20次後，因為loop的內容，使得進入條件不再成立，自動停止迴圈運算。

```
1 import UIKit
2
3 let maxLoopNum = 20
4 var loop = 0
5  {
6     loop += 1
7 }
8 print("      "+String(loop)+" loops completed")
```

20 loops completed

使用while迴圈產生等差數列，找出合法數列的最後一個數。numMax代表數列的上界，numStart代表起始數字，dis代表公差。請完成while迴圈設計，使得num代表合法數列的最後一個數

```
1 import UIKit
2
3 let numMax = 20
4 let numStart = 2
5 let dis = 3
6 var num = numStart
7  numMax {
8     num += dis
9 }
10 print("    the last number " + String(num) + " <= " + String(numMax))
```

使用 `while` 迴圈產生等差數列，找出合法數列的最後一個數。 `numMax` 代表數列的上界，`numStart` 代表起始數字，`dis` 代表公差，`sizeMax` 代表最大數列個數。請完成 `while` 迴圈設計，使得 `num` 代表合法數列的最後一個數，請使用 `&&` 設計進入條件，包括判斷 `num` 與 `numMax` 的關係，及 `size` 與 `sizeMax` 間的關係

```
1 import UIKit
2
3 let numMax = 200
4 let numStart = 2
5 let dis = 7
6 let sizeMax = 25
7 var size = 0
8 var num = numStart
9 while num < numMax && size < sizeMax {
10     size += 1
11     num += dis
12 }
13 print("size = " + String(size))
14 print(num)
```

200
2
7
25
0
2
(25 times)
(25 times)
"size = 25..."
"177\n"

```
size = 25
177
```