

### 3-2-2 以亂數產生 30 個 250~500 間不重複

```
1 #!/bin/bash
2 #Description : RANDOM TIME! 250~500
3 #Written by Rico
4 #Version 1.20
5 count=1
6 declare -a myNumber
7
8 #先判斷號碼的次數有沒有小於等於30
9 while [ $count -le 30 ]
10 do
11     #給予亂數，+1避免遇到數字0
12     let Number=$RANDOM*500/32767+1
13     #如果亂數產生低於250就使用continue回到while再跑一次，次數不會+1
14     [ $Number -lt 250 ] && continue
15     #從所有的30個數字一一比對是否有重複
16     for (( x=1 ; x<=30 ; x=x+1 ))
17     do
18         #重點在於${myNumber[$x]:-1}，功能在於一開始第一次設和
19         #後面還沒設的陣列變數，都以1代替，1也低於250所以不會重複
20         #如果數字重複的就給他一個數字，小於250也不是1即可，只要
21         #下面的continue的條件達成就可以了，不知道為何用unset有問題
22         [ $Number -eq ${myNumber[$x]:-1} ] && Number=2
23     done
24     #呼應剛剛自己設的數字，但也不要忘記數字不可以介於1~49，然後回到while
25     #的迴圈再給一次亂數
26     [ $Number -eq 2 ] && continue
27     #把產生好的數字裝在myNumber的陣列變數裡
28     myNumber[$count]=$Number
29     #確定通過上面所有的篩選才算有效給的亂數次數
30     let count=$count+1
31 done
32 #最後一次顯示所有的陣列變數
33 echo -n "${myNumber[*]} "
34 echo ""
35
36 exit 0
```

### 簡單測驗結果

```
iot@Xenial-Host:~/Rico/cc101/SS/homework$ bash 3-2-2
264 439 495 455 494 326 266 363 303 328 323 263 484 340 278 308 453 393 432 428 388 310 382 414 372 33
8 472 497 282 276 418
iot@Xenial-Host:~/Rico/cc101/SS/homework$ bash 3-2-2
468 371 401 338 291 435 354 486 396 271 368 288 388 379 284 418 433 260 414 420 335 314 324 265 469 40
9 377 410 275 389 369
iot@Xenial-Host:~/Rico/cc101/SS/homework$ bash 3-2-2
492 346 253 359 491 484 318 448 328 477 322 277 499 393 338 474 281 319 403 466 384 365 469 434 485 26
6 423 452 353 282 306
iot@Xenial-Host:~/Rico/cc101/SS/homework$ bash 3-2-2
300 491 347 282 466 499 439 297 263 446 353 269 285 278 333 341 490 488 252 364 362 478 452 497 379 25
6 400 365 307 255 464
```

```
+ '[' 29 -le 30 ']'
+ let 'Number=12324*500/32767+1'
+ '[' 189 -lt 250 ']'
+ continue
+ '[' 29 -le 30 ']'
+ let 'Number=24717*500/32767+1'
+ '[' 378 -lt 250 ']'
```

```
+ '[' 2 -le 30 ']'
+ let Number=30519*500/32767+1'
+ '[' 466 -lt 250 ']'
+ (( x=1 ))
+ (( x<=30 ))
+ '[' 466 -eq 469 ']'
+ (( x=x+1 ))
+ (( x<=30 ))
+ '[' 466 -eq 1 ']'
+ (( x=x+1 ))
+ (( x<=30 ))
+ '[' 466 -eq 1 ']'
+ (( x=x+1 ))
+ (( x<=30 ))
+ '[' 466 -eq 1 ']'
+ (( x=x+1 ))
+ (( x<=30 ))
+ '[' 466 -eq 1 ']'
+ (( x=x+1 ))
+ (( x<=30 ))
+ '[' 466 -eq 1 ']'
+ (( x=x+1 ))
+ (( x<=30 ))
```

### 3-2-4 請修改 hostChecker.sh，可以讓使用者輸入一網段，起始 IP 與結束 IP

跟原本的構想差不多，只是加入 read 可以讓使用者輸入，把變數放在相對應的就成了 IP 位址來 PING

```
1 #!/bin/bash
2 #Description : hostCheck custom
3 #Written by Rico
4 #Version 1.00
5
6 read -p "what is net you want to scan? " myNet
7 read -p "what host you want to start scan? " startIp
8 read -p "what host you want to end scan? " endIp
9
10 rightnowIp=$startIp
11
12 while [ $rightnowIp -le $endIp ]
13 do
14     echo -n "pinging $myNet.$rightnowIp....."
15     ping -c 1 -i 0.2 -W 1 $myNet.$rightnowIp &> /dev/null
16     if [ $? -eq 0 ]
17     then
18         echo "you can use that ip"
19     else
20         echo ""
21     fi
22     let rightnowIp=$rightnowIp+1
23 done
24
25 exit 0
```

PING 看看班上的人，意外還蠻多人在星期日來的(明明可以轉頭看就好)

```
iot@Xenial-Host:~/Rico/cc101/SS/homework$ bash 3-2-4
what is net you want to scan? 140.115.236
what host you want to start scan? 132
what host you want to end scan? 142
pinging 140.115.236.132.....you can use that ip
pinging 140.115.236.133.....
pinging 140.115.236.134.....you can use that ip
pinging 140.115.236.135.....you can use that ip
pinging 140.115.236.136.....
pinging 140.115.236.137.....you can use that ip
pinging 140.115.236.138.....
pinging 140.115.236.139.....
pinging 140.115.236.140.....
pinging 140.115.236.141.....
pinging 140.115.236.142.....
```

來看看運作的效果，一樣掃中央大學的網段，從 1 號(131+1)掃到最後一號(131+26)

```
iot@Xenial-Host:~/Rico/cc101/SS/homework$ bash -x 3-2-4
+ read -p 'what is net you want to scan? ' myNet
what is net you want to scan? 140.115.236
+ read -p 'what host you want to start scan? ' startIp
what host you want to start scan? 132
+ read -p 'what host you want to end scan? ' endIp
what host you want to end scan? 158
+ rightnowIp=132
+ '[' 132 -le 158 ']'
+ echo -n 'pinging 140.115.236.132.....'
pinging 140.115.236.132.....+ ping -c 1 -i 0.2 -W 1 140.115.236.132
+ '[' 1 -eq 0 ']'
+ echo
```

有開機的 ping 成功後會回傳 exit 0 剛好與判斷式吻合，就會 echo 告知使用者可用的 ip。

```
+ let rightnowIp=134+1
+ '[' 135 -le 158 ']'
+ echo -n 'pinging 140.115.236.135.....'
pinging 140.115.236.135.....+ ping -c 1 -i 0.2 -W 1 140.115.236.135
+ '[' 0 -eq 0 ']'
+ echo 'you can use that ip'
you can use that ip
```

### 3-2-6 請設計 49 選 6 大樂透的電腦選號投注程式，可自訂買一組或五組...等

```
1 #!/bin/bash
2 #Description : Y000000 you wanna rich? buy a or more lotteries
3 #Written by Rico
4 #Version 1.00
5
6 read -p "how many lottery you wanna buy? " lotteryCount
7
8 #可以依照使用者想要買幾張彩券而跑幾"行"的結果
9 for (( x=1 ; x<=$lotteryCount ; x=x+1 ))
10 do
11     #49個數字一張內雖然不可重複，但如果沒有這裡歸零的動作，如果買好幾張彩券49個數字
12     #會被陣列全部用完，所以每張彩券都要把彩券數字重設，不同張彩券數字重複沒關係
13     unset myNumber
14     declare -a myNumber
15     #開始給予每張彩券6個數字，NumberCount為第幾個數字
16     NumberCount=1
17
18     while [ $NumberCount -le 6 ]
19     do
20         #給予亂數+1，避免得到數字0
21         let Number=$RANDOM*49/32767+1
22         #開始驗證6個數字裡面是否有重複到
23         for (( z=1 ; z<=6 ; z=z+1 ))
24         do
25             #把亂數的數字跟所有6個數字去做有沒有重複的比對
26             #一定會有第一次與還沒產生的彩券號碼，所以先用一個數字代替
27             #跟上一題雷同，只是這次空值的替換要換成1~49都沒有的數字50
28             #如果數字重複的就給他一個數字，也不一定要51，只要下面的
29             #可以做continue的條件達成就可以了，不知道為何用unset有問題
30             [ $Number -eq ${myNumber[z]:-50} ] && Number=51
31         done
32         #呼應剛剛自己設的數字，但也不要忘記數字不可以介於1~49，然後回到while
33         #的迴圈再給一次亂數
34         [ $Number -eq 51 ] && continue
35         #經過重重的重複檢驗後，把數字裝進彩券的號碼(myNumber)裡
36         myNumber[$NumberCount]=$Number
37         #所有的動作都完成後才算完成產生彩券號碼
38         let NumberCount=$NumberCount+1
39     done
40     #最後顯示6個數字與換行
41     echo -n "lottery $x ) ${myNumber[@]} "
42     echo ""
43 done
44 exit 0
```

來先簡單查看全部的結果

```
iot@Xenial-Host:~/Rico/cc101/SS/homework$ bash 3-2-6
how many lottery you wanna buy? 10
lottery 1 ) 19 27 38 16 45 37
lottery 2 ) 6 33 46 25 3 21
lottery 3 ) 16 18 38 3 43 10
lottery 4 ) 8 13 49 22 17 35
lottery 5 ) 37 17 33 28 43 36
lottery 6 ) 6 14 15 18 7 2
lottery 7 ) 47 31 1 24 32 17
lottery 8 ) 30 3 24 44 8 13
lottery 9 ) 42 38 10 13 3 23
lottery 10 ) 46 21 16 11 47 12
```

看看裡面進行了什麼樣的動作，如同腳本裡所寫的，先產生幾張彩券，解除彩券號碼(myNumber)，定義彩券號碼為陣列。之後開始產生第一個號碼，然後因為是第一次產生變數，先用 1~49 都不是的號碼 50 來做重複上的驗證共 6 次，最後的 check 為若有重複時會先給予 51 的數字，然後到這裡被 continue 中斷。右半邊為最後一個彩券數字，可以看出之前骰了 21, 4, 29, 3, 2，最後的 50 其實可以不用寫，只要檢查前面 5 個數字就夠了。

```
iot@Xenial-Host:~/Rico/cc101/SS/homework$ bash -x 3-2-6
+ read -p 'how many lottery you wanna buy? ' lotteryCount
how many lottery you wanna buy? 1
+ (( x=1 ))
+ (( x<=1 ))
+ unset myNumber
+ declare -a myNumber
+ NumberCount=1
+ '[' 1 -le 6 ']'
+ let 'Number=13827*49/32767+1'
+ (( z=1 ))
+ (( z<=6 ))
+ '[' 21 -eq 50 ']' 1
+ (( z=z+1 ))
+ (( z<=6 ))
+ '[' 21 -eq 50 ']' 2
+ (( z=z+1 ))
+ (( z<=6 ))
+ '[' 21 -eq 50 ']' 3
+ (( z=z+1 ))
+ (( z<=6 ))
+ '[' 21 -eq 50 ']' 4
+ (( z=z+1 ))
+ (( z<=6 ))
+ '[' 21 -eq 50 ']' 5
+ (( z=z+1 ))
+ (( z<=6 ))
+ '[' 21 -eq 50 ']' 6
+ (( z=z+1 ))
+ (( z<=6 ))
+ '[' 21 -eq 51 ']' check
+ myNumber[$NumberCount]=21
+ let NumberCount=1+1
```

```
+ '[' 6 -le 6 ']'
+ let 'Number=9249*49/32767+1'
+ (( z=1 ))
+ (( z<=6 ))
+ '[' 14 -eq 21 ']' 1
+ (( z=z+1 ))
+ (( z<=6 ))
+ '[' 14 -eq 4 ']' 2
+ (( z=z+1 ))
+ (( z<=6 ))
+ '[' 14 -eq 29 ']' 3
+ (( z=z+1 ))
+ (( z<=6 ))
+ '[' 14 -eq 3 ']' 4
+ (( z=z+1 ))
+ (( z<=6 ))
+ '[' 14 -eq 2 ']' 5
+ (( z=z+1 ))
+ (( z<=6 ))
+ '[' 14 -eq 50 ']' 6
+ (( z=z+1 ))
+ (( z<=6 ))
+ '[' 14 -eq 51 ']' check
+ myNumber[$NumberCount]=14
+ let NumberCount=6+1
+ '[' 7 -le 6 ']'
+ echo -n 'lottery 1 ) 21' 4 29 3 2 '14 '
lottery 1 ) 21 4 29 3 2 14 + echo ''
+ (( x=x+1 ))
+ (( x<=1 ))
+ exit 0
```