

考題 1060301：迴文判斷

【試題編號】 11900-1060301

【試題名稱】 迴文判斷

【說明】 請利用『指定』迴圈控制指令，由外部資料檔讀入一個欲判斷的數字，若此數字為迴文（Palindrome，左右讀起均同，例如 12321），則印出此數字及“is apalindrome.”，若不是則印出此數字及“is not a palindrome.”

【輸入資料檔案及資料格式】 1060301.SM, 1060301.T01, 1060301.T02, 1060301.T03

- (1) 檔案型態：循序檔。
- (2) 檔案資料欄位如下：（各欄位間以逗號分隔）

	欄位 1
第一筆記錄	數字

欄位 1 數字 長整數型別 （3~9 位數的正整數）

【範例檔案】 1060301.SM

第一筆記錄	12321
-------	-------

【報表輸出】

第一題結果： 12321 is a palindrome.



解題程式碼

```
1 # -*- coding: utf-8 -*-
1 # Q1
2 import tkinter as tk

3 # 建立主視窗
4 win = tk.Tk()
5 win.geometry("800x400")
6 win.title('迴文判斷')

7 # 建立一個 LabelFrame 元件，用來顯示應檢人資料
8 group = tk.LabelFrame(win, text='應檢人資料')
9 group.pack(padx=10, pady=10)

10 # 應檢人資料
11 bdata = [
12     ["姓名", "王小美", "術科測試編號", "11101123"],
13     ["座號", "60", "考試日期", "2024-08-21"]
14 ]

15 # 顯示應檢人資料
16 for rowN in range(2):
17     for columnN in range(4):
18         t = tk.Label(group, text=bdata[rowN][columnN])
19         t.grid(row=rowN, column=columnN)

20 # 開啟檔案，讀取第一行資料
21 with open('c:/test/1060301.SM') as f:
22     data = f.readline().strip() # 去除換行符號

23 # 判斷資料是否為回文
24 if data == data[::-1]:
```

```
25     result = data + ' 是回文。'
26 else:
27     result = data + ' 不是回文。'

28 # 顯示結果
29 result = '第一題結果：' + result
30 result1_label = tk.Label(win, text=result)
31 result1_label.pack()

32 # 開始執行視窗的主迴圈
33 win.mainloop()
```



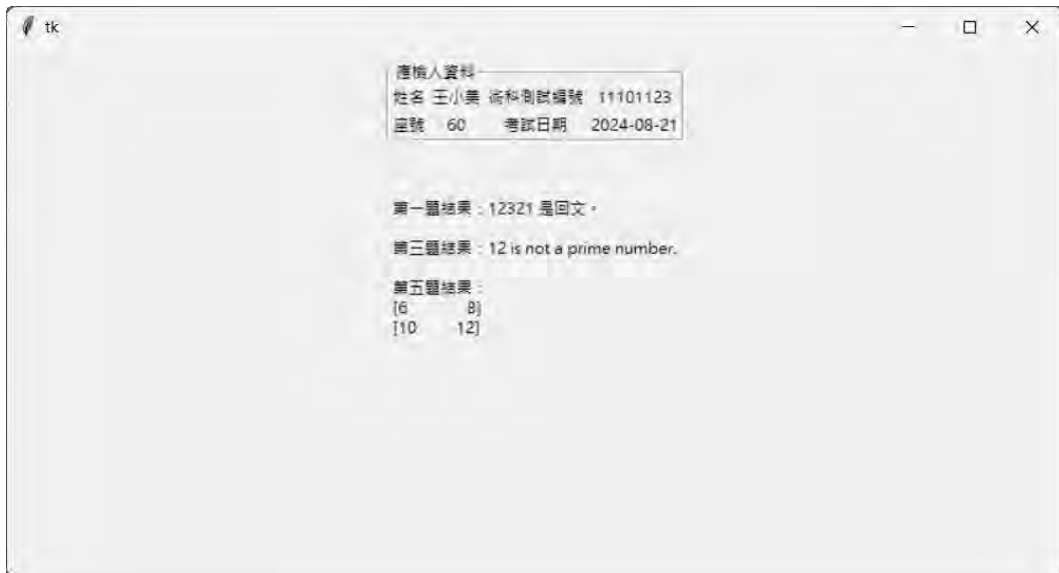
TIPS：請確認你的電腦上已經有 c:/test/1060301.SM 這個檔案，如果沒有可以自己建立一個檔案，並在第一行輸入一個字串，用來測試程式是否能正常執行。

檢定實例演習二

依題目要求把結果匯整在同一程式，同一視窗範例。

正式考試為 5 抽 3，這裡示範抽中 1、3、5 小題情形。

執行結果



完整程式碼

```
1  #import tkinter as tk
2  # 建立主視窗
3  win = tk.Tk()
4  win.geometry("800x400")
5  # 建立一個 LabelFrame 元件，用來顯示應檢人資料
6  group = tk.LabelFrame(win, text='應檢人資料')
7  group.pack(padx=10, pady=10)
8  # 應檢人資料
9  bdata = [
10     ["姓名", "王小美", "術科測試編號", "11101123"],
11     ["座號", "60", "考試日期", "2024-08-21"]
12 ]
13 # 顯示應檢人資料
14 for rowN in range(2):
15     for columnN in range(4):
16         t = tk.Label(group, text=bdata[rowN][columnN])
17         t.grid(row=rowN, column=columnN)
```

```
1  #q1
2  # 開啟檔案，讀取第一行資料
3  with open('c:/test/1060301.SM') as f:
4      data = f.readline().strip() # 去除換行符號
5  # 判斷資料是否為回文
6  if data == data[::-1]:
7      result = data + ' 是回文。'
8  else:
```

```
9     result = data + ' 不是回文。'

10  # 顯示結果
11  result1 = '第一題結果：' + result

1  #q3
2  # 讀取檔案
3  with open('c:/test/1060303.SM', encoding='utf-8') as f:
4      data = f.readline().strip() # 讀取一行文字並去除換行符號

5  # 判斷是否為質數
6  result3 = ''
7  n = int(data)
8  c = 0
9  for i in range(1,n+1):
10     if n % i == 0:
11         c += 1
12 if c == 2:
13     result3 = f'第三題結果：{data} is a prime number.'
14 else:
15     result3 = f'第三題結果：{data} is not a prime number.'

1  #q5
2  # 讀取檔案資料並轉成二維串列
3  with open('c:/test/1060305.SM', encoding='utf-8') as f:
4      datas = f.readlines()
5  d = [i.strip().split(',') for i in datas]
6  d = [[int(x) for x in sublist] for sublist in d]

7  # 計算 c 串列的值
8  c = [[0,0],[0,0]]
9  c[0][0] = d[0][0] + d[2][0]
10 c[0][1] = d[0][1] + d[2][1]
11 c[1][0] = d[1][0] + d[3][0]
```

```
12 c[1][1] = d[1][1] + d[3][1]

13 # 格式化 r 串列的值
14 r = []
15 for i in c:
16     if i[1]<10:
17         r.append( '['+str(i[0])+ ' '*15+str(i[1])+']')
18     else:
19         r.append( '['+str(i[0])+ ' '*10+str(i[1])+']')
20 result5 = '\n'.join(r)
```

```
21 result5 = '第五題結果：\n'+result5
```

```
22 resultall = ''
23 resultall = resultall +'\n\n'+ result1
24 resultall = resultall +'\n\n'+ result3
25 resultall = resultall + '\n\n' + result5
```

```
26 resultall_label = tk.Label(win, text=resultall, anchor="e",
    justify='left')
27 resultall_label.pack()

28 win.mainloop()
```

考題 1060307：撲克牌比大小

【試題編號】11900-1060307

【試題名稱】撲克牌比大小

【說明】使用一副撲克牌進行多次之發牌及比牌程序，每次各發一張牌給莊家與玩家，再按照撲克牌之大小比牌決定雙方之輸贏，撲克牌之大小依序為 $A > K > Q > J > 10 > 9 > 8 > 7 > 6 > 5 > 4 > 3 > 2$ ，不分花色，若兩張相同點數但不同花色，則為平手。發牌時以模擬隨機之機制來發牌，其做法為將數個（不大於 100 個）大於等於 0 且小於 1 之隨機數儲存於輸入檔中，當欲發牌時再由程式從輸入檔中讀出所儲存之隨機數做為發牌使用。輸入檔第 1 筆資料為發牌與比牌重覆進行之次數，第 2 筆資料之後為隨機數。下圖為一副撲克牌按其花色及大小排列並逐一編號，其中數字即為各張牌之編號，而將一隨機數 $\times 52$ ，並捨棄小數部份後所得之整數即可對應到圖中 52 張牌中之其中一張牌，其即為發牌之程序，但須注意同一張牌不可發出二次或二次以上，遇取得之牌張已發出時，則捨棄該牌張，重新另取一牌張。發牌完成後即進行比牌，按前述之撲克牌之大小比牌，較大者為贏家，若雙方點數相同則為平手。輸出範例之畫面上每一橫列即為每次發牌及比牌之結果，序號為每一橫列之編號，代表第幾次之發牌及比牌動作，莊家及玩家兩欄分別顯示莊家及玩家所持牌張之花色及點數，比牌結果則輸出莊家及玩家持牌之大小，若雙方點數相同則顯示『平手』，若莊家點數比較高則顯示『莊家贏』，反之，若玩家點數比較高則顯示『玩家贏』。畫面中花色符號之圖案其 UTF-8 編碼如下：

- (1) 花色♠={226, 153, 160}
- (2) 花色♥={226, 153, 165}
- (3) 花色♦={226, 153, 166}
- (4) 花色♣={226, 153, 163}

下列為產生各花色之程式碼，本程式碼僅供參考，最後繳交之程式碼不以此為限。

```
Dim suit(4) As String
Dim ba1() As Byte = {226, 153, 160}
Dim ba2() As Byte = {226, 153, 165}
Dim ba3() As Byte = {226, 153, 166}
Dim ba4() As Byte = {226, 153, 163}
suit(0) = Encoding.UTF8.GetString(ba1)
suit(1) = Encoding.UTF8.GetString(ba2)
suit(2) = Encoding.UTF8.GetString(ba3)
suit(3) = Encoding.UTF8.GetString(ba4)
```



【功能動作要求】

- (1) 程式執行時需按範例畫面與「壹、試題使用說明」第五、六兩項規定設計。程式製作時，先以範例資料檔案 1060307.SM 進行測試。若結果與螢幕輸出範例相同時，再以測試檔案 1060307.T01 為輸入檔案完成受測。

測試檔案的筆數不同於範例資料檔案

測試檔案型態格式和範例資料檔案相同

- (2) 讀取輸入檔第 1 筆資料以決定發牌及比牌進行之次數。
- (3) 每次莊家及玩家均以模擬隨機之機制各發 1 張牌，並將各家之持牌及點數分別顯示在莊家及玩家之欄位，發牌完成後進行比牌，比牌結果顯示在結果欄位，若雙方點數相同則顯示『平手』，若莊家點數比較高則顯示『莊家贏』，反之，若玩家點數比較高則顯示『玩家贏』（請參考【輸出範例】）。
- (4) 重覆前述第 3 點之動作，直到進行之次數已達第 2 點所讀取到之次數，而其顯示之位置由最初第 1 列開始，隨後每次下移一列。每列之序號欄位為顯示第幾次之發牌及比牌動作。
- (5) 將程式連同輸出結果，列印於報表上，在報表右上角簽名，等評審完畢後繳交。



解題程式碼

```
1 # 模組載入
2 from tkinter import *
3 from tkinter import ttk
4 import tkinter as tk

5 # 建立視窗
6 win = Tk()
7 win.geometry("800x400")
8 win.title('撲克牌比大小')

9 # 建立應檢人資料框架
10 group = tk.LabelFrame(win, text='應檢人資料')
11 group.pack(padx=10, pady=10)
12
13 # 建立應檢人資料表格
14 bdata = [
15     ["姓名", "王小美", "術科測試編號", "11101123"],
16     ["座號", "60", "考試日期", "2022-03-05"]
17 ]
18
19 for rowN in range(2):
20     for columnN in range(4):
21         if columnN not in [1, 3]:
22             t = Label(group, text=bdata[rowN][columnN])
23         else:
24             t = tk.Entry(group, width=10)
25             t.insert(0, bdata[rowN][columnN])
26         t.grid(row=rowN, column=columnN)
```

```
27 # 讀取資料
28 with open('c:/test/1060307.SM') as f:
29     datas = f.readlines()

30 n = int(datas[0])
31 rd = []
32 for i in range(1, len(datas)):
33     t = int(float(datas[i]) * 52)
34     rd.append(t)

35 # 去重複、計算比大小
36 gd = []
37 rd1 = []
38 for i in rd:
39     if i not in rd1:
40         rd1.append(i)
41 j = 1
42 for i in range(0, n * 2, 2):
43     p = rd1[i]           # p: 玩家
44     pn = p % 13 + 1     # pn: 玩家牌數字
45     pf = p // 13       # pf: 玩家牌花色
46     b = rd1[i + 1]     # b: 莊家
47     bn = b % 13 + 1   # bn: 莊家牌數字
48     bf = b // 13      # bf: 莊家牌花色
49     msg = ''
50     if pn > bn:
51         msg = '玩家贏'
52     if pn == bn:
53         msg = '平手'
54     if pn < bn:
55         msg = '莊家贏'
56     rn = ['', 'A', '2', '3', '4', '5', '6', '7', '8', '9', '10',
            'J', 'Q', 'K']
```

```
57     rf = ['\u2660', '\u2665', '\u2666', '\u2663']
      # 牌花色 utf-8 碼
58     t = [j, rf[pf] + rn[pn], rf[bf] + rn[bn], msg]
59     gd.append(t)
60     j = j + 1

61 # 建立結果表格
62 hname = ["序號", "玩家", "莊家", "結果"]
63 tree = ttk.Treeview(win, column=hname, show='headings')

64 for i in hname:
65     tree.column(i, anchor=W)
66     tree.heading(i, text=i, anchor=W)

67 for i in range(len(gd)):
68     tree.insert('', 'end', text="1", values=gd[i])

69 tree.pack()

70 # 執行主程式
71 win.mainloop()
```