


# ĐỀ THI PYTHON CƠ BẢN

**Thời gian: 120 phút Ngày thi: 18/07/2022 - Được sử dụng tài liệu.**

- Khi nộp bài, học viên gửi các file bài làm của mình (chỉ gửi các file .py) đến email [levanhanhvn@gmail.com](mailto:levanhanhvn@gmail.com).

**CÂU 1. (1 điểm)** Cho nhập 1 số nguyên  $n$  ( $n > 1$ ), nếu nhập sai, yêu cầu nhập lại cho đến khi nhập đúng. Sau khi nhập đúng, chương trình in ra chuỗi số Fibonacci gồm  $n$  số.

Ví dụ: với  $n = 10$ , sẽ in ra 0 1 1 2 3 5 8 13 21 34


 **Nhắc lại:** chuỗi số Fibonacci luôn bắt đầu từ 2 số 0 và 1. Mỗi số tiếp theo được xác định bằng cách cộng hai số liền trước nó.

**CÂU 2. (3 điểm)** Lần lượt cài đặt các hàm có trong yêu cầu, sau đó viết chương trình gọi các hàm này thực hiện:

**a. (1 điểm)** Viết hàm cho người dùng nhập số nguyên dương  $n$  có giá trị nằm trong khoảng từ 10 đến 1000. Nếu nhập đúng, hàm tạo ra 1 list chứa  $n$  các giá trị ngẫu nhiên không trùng nhau nằm trong khoảng từ 1 đến 2000 và hàm trả về list vừa tạo được. Ngược lại nếu nhập sai, chương trình sẽ yêu cầu nhập lại cho đến khi nhập đúng. Các trường hợp được xem là  $n$  nhập sai:

- (i)- Nếu người dùng nhập kiểu dữ liệu không phải là số nguyên (như string, float, bool, ...) chương trình sẽ hiện ra câu báo lỗi '*Phải nhập số nguyên dương. Yêu cầu nhập lại*'.
- (ii)- Nếu giá trị nhập vào của  $n$  đúng kiểu dữ liệu nguyên nhưng giá trị không nằm trong khoảng 10 đến 1000, chương trình báo lỗi '*Chỉ nhận giá trị từ 10 đến 1000. Yêu cầu nhập lại*'.

**b. (1 điểm)** Lucky Number

 **Lucky Number:** (Theo Wikipedia) *Lucky Number (số may mắn)* là số được định nghĩa theo quá trình sau: bắt đầu với số nguyên dương  $x$  và tính tổng bình phương  $y$  các chữ số của  $x$ , sau đó tiếp tục tính tổng bình phương các chữ số của  $y$ . Quá trình này lặp đi lặp lại cho đến khi thu được kết quả là 1 thì dừng (tổng bình phương các chữ số của số 1 chính là 1) hoặc quá trình sẽ kéo dài vô tận. Số mà quá trình tính này kết thúc bằng 1 gọi là số may mắn. Số có quá trình tính kéo dài vô tận là *số không may mắn* hay còn gọi là *sad number (số Đen Đủi)*

- Ví dụ: 7 là số may mắn vì:

$$7^2 = 49$$


$$4^2 + 9^2 = 97$$

$$9^2 + 7^2 = 130$$

$$1^2 + 3^2 + 0^2 = 10$$

$$1^2 + 0^2 = 1$$

- Minh họa những số may mắn dưới 500 là: 1, 7, 10, 13, 19, 23, 28, 31, 32, 44, 49, 68, 70, 79, 82, 86, 91, 94, 97, 100, 103, 109, 129, 130, 133, 139, 167, 176, 188, 190, 192, 193, 203, 208, 219, 226, 230, 236, 239, 262, 263, 280, 291, 293, 301, 302, 310, 313, 319, 320, 326, 329, 331, 338, 356, 362, 365, 367, 368, 376, 379, 383, 386, 391, 392, 397, 404, 409, 440, 446, 464, 469, 478, 487, 490, 496.

 **Yêu cầu cài đặt:** Viết hàm liệt kê các số may mắn có trong list vừa tạo ở câu a.

Giả sử để xác định quá trình tìm *Lucky number* có kéo dài vô tận hay không, người ta cho lặp việc tìm này tối đa 100 lần. Do đó nếu quá trình lặp để tìm vượt trên 100 lần thì chương trình có thể kết luận số đó không là *Lucky number*.

➤ Kết quả gợi ý khi thực hiện chương trình:

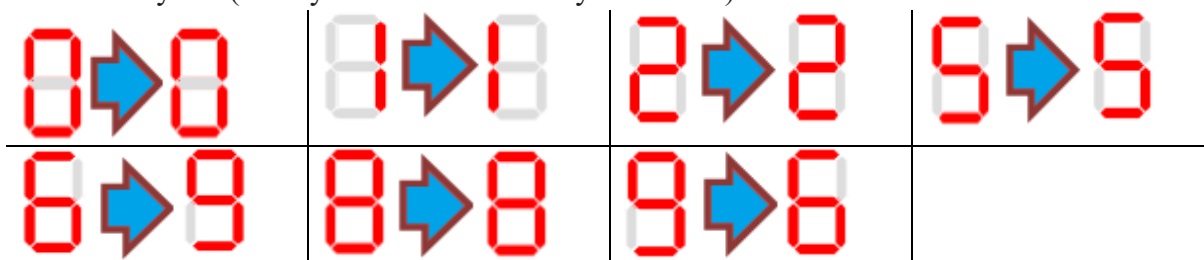
- Nếu trong list không có số may mắn nào thì in ra thông báo:  
*Trong list không chứa số may mắn.*
- Nếu trong list có số may mắn thì in ra thông báo:  
*Các số may mắn có trong list là ...*

c. (1 điểm) Số nguyên tố *strobogrammatic*

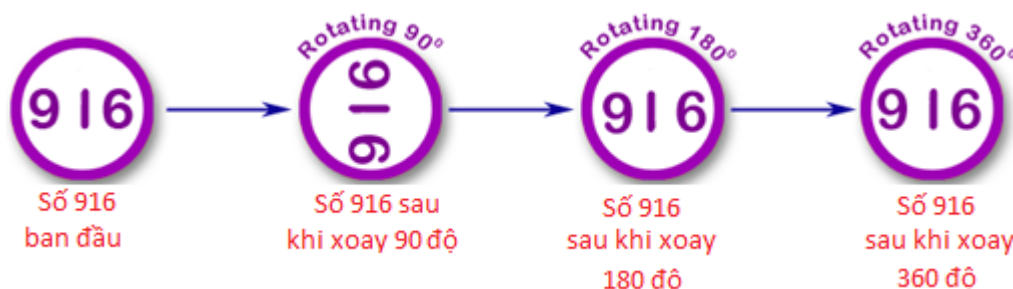
Giả sử có định nghĩa về số *strobogrammatic*:

📖 Số *strobogrammatic*

- Là một số có giá trị không đổi khi xoay số đó 180 độ (180°).
- Khi xoay các số dạng digital, ta thấy các số sau đây vẫn có ý nghĩa là 0, 1, 2, 5, 6, 8, 9. Trong đó các số 0, 1, 2, 5, 8 không bị thay đổi giá trị sau khi xoay, còn số 6 và 9 bị thay đổi (6 chuyển thành 9 và 9 chuyển thành 6)



- Ví dụ 1: số 68910 sau khi xoay sẽ là 01689  $\Rightarrow$  68910 không phải là số *Strobogrammatic* vì  $68910 \neq 01689$
- Ví dụ 2:



➤ Yêu cầu cài đặt: Viết hàm liệt kê các số vừa là số nguyên tố, vừa là số *strobogrammatic* có trong list vừa tạo ở câu a.

➤ Kết quả gợi ý khi thực hiện chương trình: tương tự gợi ý khi thực hiện chương trình ở câu b.

**CÂU 3. (1 điểm)** Liệt kê ra các số lộc phát nhỏ hơn 10000. Với yêu cầu chỉ kết hợp hàm lambda (Anonymous Function) và cú pháp Comprehension để thực hiện, do đó sẽ không được gọi các hàm được định nghĩa bằng từ khóa def.

📖 Số Lộc Phát: giả sử người ta cho rằng 1 số gọi là số Lộc Phát nếu chỉ chứa toàn các số 6 hoặc số 8.

Ví dụ:  $n=686 \Rightarrow 686$  là số Lộc Phát.

$n=68626 \Rightarrow 68626$  KHÔNG phải số Lộc Phát.

**CÂU 4. (1 điểm)** Sử dụng cú pháp comprehension để tạo ra list L chứa các sublist (list con bên trong L) sao cho mỗi sublist chứa 3 giá trị nguyên trong khoảng từ 1 đến 100 và bình phương của 1 số trong đó bằng tổng bình phương của 2 số còn lại.

- Kết quả thực hiện đúng list L sẽ gồm 52 sublist, cụ thể là: `[3, 4, 5]`, `[5, 12, 13]`, `[6, 8, 10]`, `[7, 24, 25]`, `[8, 15, 17]`, `[9, 12, 15]`, `[9, 40, 41]`, `[10, 24, 26]`, `[11, 60, 61]`, `[12, 16, 20]`, `[12, 35, 37]`, `[13, 84, 85]`, `[14, 48, 50]`, `[15, 20, 25]`, `[15, 36, 39]`, `[16, 30, 34]`, `[16, 63, 65]`, `[18, 24, 30]`, `[18, 80, 82]`, `[20, 21, 29]`, `[20, 48, 52]`, `[21, 28, 35]`, `[21, 72, 75]`, `[24, 32, 40]`, `[24, 45, 51]`, `[24, 70, 74]`, `[25, 60, 65]`, `[27, 36, 45]`, `[28, 45, 53]`, `[28, 96, 100]`, `[30, 40, 50]`, `[30, 72, 78]`, `[32, 60, 68]`, `[33, 44, 55]`, `[33, 56, 65]`, `[35, 84, 91]`, `[36, 48, 60]`, `[36, 77, 85]`, `[39, 52, 65]`, `[39, 80, 89]`, `[40, 42, 58]`, `[40, 75, 85]`, `[42, 56, 70]`, `[45, 60, 75]`, `[48, 55, 73]`, `[48, 64, 80]`, `[51, 68, 85]`, `[54, 72, 90]`, `[57, 76, 95]`, `[60, 63, 87]`, `[60, 80, 100]`, `[65, 72, 97]`
- Lưu ý: cần loại bỏ các sublist mà thành phần bên trong chúng là giống nhau và chỉ khác nhau do đảo vị trí. Ví dụ: Chỉ giữ lại 1 trong các sublist sau trong kết quả: `[3, 4, 5]`, `[4, 3, 5]`, `[5, 3, 4]`, ...

**CÂU 5. (1 điểm)**

Tạo 2 listA, listB với kích thước khác nhau và chứa giá trị tùy ý. Ví dụ:

```
listA = [1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89]
listB = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13]
```

Viết chương trình trả về *listAinB* cho biết những giá trị nào trong *listA* có trong *listB* và ngược lại tạo 1 *listBinA* chứa những giá trị trong listB có trong listA.

➤ Yêu cầu lần lượt thực hiện bằng 2 cách sau:

- (0.5 điểm) Sử dụng Comprehension
- (0.5 điểm) Sử dụng Anonymous Function

**CÂU 6. (3 điểm) Viết chương trình đọc và ghi file CSV**

Đầu giờ thi, GV sẽ cung cấp file *DiaLyVietNam.csv* để học viên làm bài. Nội dung file này cho biết thông tin về các tỉnh thành phố tại Việt Nam.

➤ *Yêu cầu chung*: phải sử dụng lệnh *try ... except* cho tất cả các câu sau đây để kiểm soát các lỗi xảy ra nếu có.

➤ *Yêu cầu thực hiện*:

- (2đ) Viết hàm *InNoiDungFile\_CSV(filename)* để đọc và in ra nội dung của những dòng trong tập tin csv (được truyền vào thông qua tham số *filename*) thỏa điều kiện ra màn hình.  
Yêu cầu cài đặt: chọn ra những tỉnh - thành phố có biên giới với 2 nước khác nhau hoặc vừa có biên giới với 1 nước, vừa giáp với biển

Kết quả in dữ liệu ra màn hình theo dạng sau:

**DANH SÁCH CÁC TỈNH THÀNH PHỐ**

**CÓ BIÊN GIỚI VỚI 2 NƯỚC HOẶC VỪA CÓ BIÊN GIỚI VỚI 1 NƯỚC, VỪA CÓ GIÁP BIỂN**

TT	Khu vực	Tỉnh-Thành phố	Dân số	Diện tích	BG với TQ	BG với Lào	BG với Campuchia	Giáp biển
1	Bắc Trung Bộ	Hà Tĩnh	1288866	5990.7	Không	Có	Không	Có
2	Bắc Trung Bộ	Nghệ An	3327791	16493.7	Không	Có	Không	Có
3	Bắc Trung Bộ	Quảng Bình	895430	8065.3	Không	Có	Không	Có
4	Bắc Trung Bộ	Quảng Trị	632375	4739.8	Không	Có	Không	Có
5	Bắc Trung Bộ	Thanh Hóa	3640128	11114.7	Không	Có	Không	Có
6	Bắc Trung Bộ	Thừa Thiên Huế	1128620	5048.2	Không	Có	Không	Có
7	Đông Bắc Bộ	Quảng Ninh	1320324	6177.7	Có	Không	Không	Có
8	Đồng bằng sông Cửu Long	Kiên Giang	1723067	6348.8	Không	Không	Có	Có
9	Duyên hải Nam Trung Bộ	Quảng Nam	1495812	10574.7	Không	Có	Không	Có
10	Tây Bắc Bộ	Điện Biên	598856	9541	Có	Có	Không	Không
11	Tây Nguyên	Kon Tum	540438	9674.2	Không	Có	Có	Không

**Tổng cộng gồm 11 Tỉnh-Thành phố, với tổng diện tích là 16591707 và tổng dân số là 93768.8**

- b.** (1đ) Viết hàm *GhiFile\_CSV (filename, listname)* để ghi nội dung list vừa có ở câu a vào vào tập tin csv được khai báo thông qua filename, với dấu phân cách (delimiter) là dấu phẩy(.). Nội dung của file kết quả (ketqua.csv) như sau:

Tên khu vực,Tên Tỉnh-Thành phố,Dân số,Diện tích,Có biên giới với TQ,Có biên giới với Lào,Có biên giới với Campuchia,Có giáp với biển  
 Bắc Trung Bộ,Hà Tĩnh,1288866,5990.7,Không,Có,Không,Có  
 Bắc Trung Bộ,Nghệ An,3327791,16493.7,Không,Có,Không,Có  
 Bắc Trung Bộ,Quảng Bình,895430,8065.3,Không,Có,Không,Có  
 Bắc Trung Bộ,Quảng Trị,632375,4739.8,Không,Có,Không,Có  
 Bắc Trung Bộ,Thanh Hóa,3640128,11114.7,Không,Có,Không,Có  
 Bắc Trung Bộ,Thừa Thiên Huế,1128620,5048.2,Không,Có,Không,Có  
 Đông Bắc Bộ,Quảng Ninh,1320324,6177.7,Có,Không,Không,Có  
 Đồng bằng sông Cửu Long,Kiên Giang,1723067,6348.8,Không,Không,Có,Có  
 Duyên hải Nam Trung Bộ,Quảng Nam,1495812,10574.7,Không,Có,Không,Có  
 Tây Bắc Bộ,Điện Biên,598856,9541,Có,Có,Không,Không  
 Tây Nguyên,Kon Tum,540438,9674.2,Không,Có,Có,Không

**--- Chúc các bạn làm bài tốt ---**