

CODE MẪU BÀI TẬP TUẦN 8

I. Phần cơ bản:

Câu 1: Anagram

Anagram là thuật ngữ để chỉ hai chuỗi có số ký tự và các ký tự giống nhau, nhưng thứ tự của các ký tự trong chuỗi có thể khác nhau. Viết chương trình cho phép nhập từ bàn phím hai chuỗi a và b và kiểm tra xem hai chuỗi này có phải là anagram của nhau hay không.

Input:

- Dòng 1: chuỗi a
- Dòng 2: chuỗi b

Output:

- Dòng 1: "co" nếu hai chuỗi là anagram, "khong" nếu hai chuỗi không là anagram

Ví dụ:

Input	Output
listen sliten	co
sliten list	khong

Code:

```
def anagram(str1, str2):  
    if len(str1) != len(str2): #Kiểm tra 2 xâu có cùng số lượng kí tự  
hay không  
        return False  
  
    #Kiểm tra xem 2 xâu đã được sắp xếp có giống nhau hay không  
    if sorted(str1) == sorted(str2):  
        return True  
    else:  
        return False  
  
a = input()  
b = input()  
if anagram(a, b): #Kiểm tra xem 2 xâu có là anagram hay không  
    print("co")  
else:  
    print("khong")
```

Câu 2: Biển số xe

Biển số xe là một chuỗi gồm 11 kí tự trong đó 2 kí tự đầu là 2 chữ số, tiếp theo là dấu gạch ngang, 1 kí tự tiếp theo là một chữ cái, 4 kí tự tiếp theo là các chữ số, tiếp theo là dấu chấm và cuối cùng là 2 chữ số. Viết chương trình cho phép nhập từ bàn phím một chuỗi s, kiểm tra xem đây có phải là một biển số xe hay không?

Input:

- Dòng 1: chuỗi s

Output:

- Dòng 1: "Yes" nếu là một chuỗi biển số. "No" nếu ngược lại

Ví dụ:

Input	Output
99-B2234.56	Yes
99 - B2 686.66	No
99-12345.B6	No

Code:

```
def biensoxe(str):
    if len(str) != 11: #Kiểm tra độ dài xâu
        return False
    if not str[:2].isdigit(): #Kiểm tra ký tự đầu là 2 chữ số
        return False
    if str[2] != "-": #Kiểm tra ký tự tiếp theo là dấu gạch ngang
        return False
    if not str[3].isalpha(): #Kiểm tra ký tự tiếp theo là một chữ cái
        return False
    if not str[4:8].isdigit(): #Kiểm tra 4 ký tự tiếp theo là chữ số
        return False
    if str[8] != ".": #Kiểm tra ký tự tiếp theo là dấu chấm
        return False
    if not str[9:].isdigit(): #Kiểm tra 2 ký tự cuối là 2 chữ số
        return False
    return True

a = input()
if biensoxe(a):
    print("Yes")
else:
    print("No")
```

Câu 3: Biển số xe Hà Nội

Biển số xe là một chuỗi gồm 11 kí tự trong đó 2 kí tự đầu là 2 chữ số, tiếp theo là dấu gạch ngang, 1 kí tự tiếp theo là một chữ cái, 4 kí tự tiếp theo là các chữ số, tiếp theo là dấu chấm và cuối cùng là 2 chữ số. Viết chương trình cho phép nhập từ bàn phím một chuỗi s, kiểm tra xem đây có phải là một biển số xe Hà nội hay không? (Biển số xe Hà nội là biển số bắt đầu bằng kí tự "29" hoặc "30" hoặc "31")

Input:

- Dòng 1: chuỗi s

Output:

- Dòng 1: "Yes" nếu là một chuỗi biển số xe Hà nội. "No" nếu ngược lại

Ví dụ:

Input	Output
29-B2234.56	Yes
99-C2686.66	No
30-12345.B6	No

Code:

```
def biensoxeHN(str):
    if len(str) != 11: #Kiểm tra độ dài xâu
        return False
    if str[:2] not in ['29', '30', '31']: #Kiểm tra 2 ký tự đầu có
        #phải ký tự 29 hoặc 30 hoặc 31 hay không
        return False
    if str[2] != "-": #Kiểm tra ký tự tiếp theo là dấu gạch ngang
        return False
    if not str[3].isalpha(): #Kiểm tra ký tự tiếp theo là một chữ cái
        return False
    if not str[4:8].isdigit(): #Kiểm tra 4 ký tự tiếp theo là chữ số
        return False
    if str[8] != ".": #Kiểm tra ký tự tiếp theo là dấu chấm
        return False
    if not str[9:].isdigit(): #Kiểm tra 2 ký tự cuối là 2 chữ số
        return False
    return True

a = input()
if biensoxeHN(a):
    print("Yes")
else:
    print("No")
```

Câu 4: Chuỗi đối xứng

Chuỗi đối xứng là chuỗi viết từ trái sang phải hoặc từ phải sang trái đều như nhau. Viết chương trình cho phép thực hiện các công việc sau:

- Nhập vào một chuỗi s
- Trả về "Yes" nếu chuỗi là đối xứng, ngược lại là "No"

Input:

- Dòng 1: chuỗi s

Output:

- Dòng 1: Chữ "Yes" hoặc "No" tương ứng.

Ví dụ:

Input	Output
madam	Yes
hello	No

Code:

```
s = input()
str = s[::-1] #Xâu str là xâu viết từ phải sang trái của xâu s
if s == str:
    print('Yes')
else:
    print('No')
```

Câu 5: Lấy họ trong tên

Viết chương trình cho phép nhập vào từ bàn phím một họ tên nào đó của một người Việt Nam. Hiển thị ra họ của người đó. (Họ là từ đầu tiên trong họ tên)

Input:

- Dòng 1: họ tên

Output:

- Dòng 1: tên

Ví dụ:

Input	Output
Dang Thanh Trung	Dang
nguyen thi thanh huyen	nguyen

Code:

```
ho_ten = input().rstrip() #Nhập họ tên từ bàn phím và xóa các khoảng
trắng ở đầu xâu
vt = ho_ten.find(" ") #Tìm vị trí của khoảng trắng đầu tiên trong xâu
ho = ho_ten[:vt+1] #Tách họ
print(ho)
```

Câu 6: Lấy tên trong họ tên

Viết chương trình cho phép nhập vào từ bàn phím một họ tên nào đó của một người Việt Nam. Hiển thị ra tên của người đó. (Tên là từ cuối cùng trong họ tên)

Input:

- Dòng 1: họ tên

Output:

- Dòng 1: tên

Ví dụ:

Input	Output
Dang Thanh Trung	Trung
nguyen thi thanh huyen	huyen

Code:

```
ho_ten = input().rstrip() #Nhập họ tên từ bàn phím và xóa các khoảng
trắng cuối cùng của xâu
vt = ho_ten.rfind(" ") #Tìm vị trí của khoảng trắng cuối cùng trong
xâu
ten = ho_ten[vt+1:] #Tách tên
print(ten)
```

Câu 7: Viết hoa và viết thường ký tự

Viết chương trình cho phép nhập vào từ bàn phím một chuỗi ký tự bất kỳ s từ bàn phím và thực hiện các công việc sau:

- Viết thường tất cả các kí tự trong s
- Viết hoa tất cả các kí tự trong s

Input:

- Dòng 1: chuỗi s

Output:

- Dòng 1: chứa chuỗi s viết thường
- Dòng 2: chứa chuỗi s viết hoa

Ví dụ:

Input	Output
Cntt - DH SP HN	cntt - dh sp hn
	CNTT - DH SP HN

Code:

```
s = input()
s1 = s.lower() #Viết thường tất cả các kí tự trong xâu s1
s2 = s.upper() #In hoa tất cả các kí tự trong xâu s2
print(s1)
print(s2)
```

Câu 8: Viết hoa đầu từ

Viết chương trình cho phép nhập vào từ bàn phím một chuỗi ký tự bất kỳ s từ bàn phím và thực hiện các công việc sau:

- Viết hoa các ký tự đầu từ trong s và viết thường các ký tự còn lại

Input:

- Dòng 1: chuỗi s

Output:

- Dòng 1: chứa chuỗi s đã xử lý

Ví dụ:

Input	Output
Cntt - DH SPHN	Cntt - Dh Sphn

Code:

```
s = input()
s1 = s.title() #In hoa các ký tự đầu của các từ và viết thường các ký
tự còn lại trong xâu s1
print(s1)
```


Câu 9: Đảo ngược chuỗi

Viết chương trình cho phép nhập từ bàn phím hai chuỗi s và hiển thị ra màn hình chuỗi đảo ngược của s.

Input:

- Dòng 1: chuỗi s

Output:

- Dòng 1: chuỗi đảo ngược của s

Ví dụ:

Input	Output
listen	netsil
list	tsil

Code:

```
s = input()
s1 = s[::-1] #Xâu s1 là xâu đảo ngược của s
print(s1)
```

Câu 10: Viết thường đầu từ

Viết chương trình cho phép nhập vào từ bàn phím một chuỗi ký tự bất kỳ s từ bàn phím và thực hiện các công việc sau:

- Viết thường các ký tự đầu từ trong s và viết hoa các ký tự còn lại

Input:

- Dòng 1: chuỗi s

Output:

- Dòng 1: chứa chuỗi s đã xử lý

Ví dụ:

Input	Output
Cnnt - DH SPHN	cNTT - dH sPHN

Code:

```
s = input()
s1 = s.title() #In hoa các ký tự đầu của các từ và viết thường các ký
tự còn lại trong xâu s1
s2 = s1.swapcase() #Kiểu chữ viết của tất cả các ký tự trong s2 được
đảo ngược (thường --> hoa, hoa--> thường)
print(s3)
```

Câu 11: Đếm số chuỗi xuất hiện trong chuỗi

Viết chương trình cho phép nhập từ bàn phím hai chuỗi a và b và thực hiện các công việc sau:

- Đếm số lần xuất hiện của chuỗi b trong a

Input:

- Dòng 1: chuỗi a
- Dòng 2: chuỗi b

Output:

- Dòng 1: Số lần xuất hiện của b trong a

Ví dụ:

Input	Output
what you did not what you see what	2
tomorrow never die yesterday	0

Code:

```
a = input()
b = input()
dem = 0
#Ý tưởng: duyệt từ đầu đến len(a) - len(b)
#Tại mỗi vòng lặp, kiểm tra xâu con s[vị trí i đến vị trí (i+len(b)]
#Nếu xâu con s == xâu b thì tăng biến đếm lên 1 đơn vị
for i in range(len(a)-len(b)+1):
    s = a[i:i+len(b)]
    if s==b:
        dem +=1
print(dem)
```

Câu 12: Đếm số từ trong câu

Viết chương trình cho phép nhập vào từ bàn phím một chuỗi ký tự bất kỳ s từ bàn phím và đếm số từ trong câu (từ là một chuỗi các ký tự liên tiếp nhau, không có dấu cách)

Input:

- Dòng 1: chuỗi s

Output:

- Dòng 1: Số từ đếm được

Ví dụ:

Input	Output
Cntt - DH SP HN	5
Tomorrow never die	3

Code:

```
s = input().strip(' ') #Loại bỏ các ký tự khoảng trắng ở đầu và cuối
xâu
dem = 0
if s==' ': dem=0
elif s[0] !=' ':
    dem += 1
for i in range(len(s)): #Số từ = số khoảng cách + 1
    if (s[i]==' ' and s[i+1]!=' '):
        dem += 1
print(dem)
```

II. Phần nâng cao:

Câu 1: Đổi định dạng date

Hãy viết chương trình cho phép đổi định dạng ngày tháng “yyyy-mm-dd” thành định dạng “dd-mm-yyyy”

Input: một chuỗi kí tự định dạng “yyyy-mm-dd” trong đó yyyy, mm, dd là năm, tháng, ngày hợp lệ

Output: định dạng ngày tháng “yyyy-mm-dd” được đổi thành định dạng “dd-mm-yyyy”

For example:

Input	Output
1982-09-20	20-09-1982

Code:

```
s = input()
nam, thang, ngay = s.split('-')
print(ngay + '-' + thang + '-' + nam)
```

Câu 2: Định dạng kiểu Mỹ

Viết chương trình chuyển đổi một số thực ở định dạng kiểu Mỹ (phân tách hàng nghìn bằng dấu phẩy, phân tách phần nguyên với phần thập phân bằng dấu chấm) sang định dạng kiểu Việt Nam (phân tách hàng nghìn bằng dấu chấm, phân tách phần nguyên với phần thập phân bằng dấu phẩy).

Input: một số thực ở định dạng kiểu Mỹ

Output: số thực kiểu Mỹ chuyển sang định dạng kiểu Việt Nam

For example:

Input	Output
1,234.567	1.234,567

Code:

```
so = input()
a = so.split(',')
a[-1] = a[-1].replace('.', ',')
so_moi = '.'.join(a)
print(so_moi)
```



Câu 3: Chuyển hệ cơ số

Hãy viết chương trình cho phép thực hiện việc chuyển hệ cơ số như sau:

1. Chuyển một số hệ thập phân thành hệ nhị phân
2. Chuyển một số hệ thập phân thành hệ bát phân
3. Chuyển đổi một số hệ thập phân thành hệ thập lục phân

Input: Một số hệ thập phân

Output

- Dòng 1: Số hệ thập phân được chuyển thành hệ nhị phân
- Dòng 2: Số hệ thập phân được chuyển thành hệ bát phân
- Dòng 3: Số hệ thập phân được chuyển thành hệ thập lục phân

```
#Code:
'''
    Thuật toán để chuyển một số thập phân (number) thành các
    số thuộc hệ 2, 8 và 16 như sau:
    - B1: Chia lấy phần dư number cho các cơ số tương ứng (2,
    8, 16)
    - B2: Lưu trữ lại phần dư của B1 vào 1 xâu (phép cộng
    xâu)
    - B3: Chia lấy phần nguyên number cho các cơ số tương ứng
    (2, 8, 16)
    - B4: Lặp lại B1 cho đến khi number = 0
    - B5: Viết ngược lại xâu lưu trữ được ở B2

    Lưu ý với hệ Hexa (16): Nếu B1 trả về kết quả dư từ 10-
    15, ta sẽ cộng vào xâu kết quả các chữ cái t/ư:
    - Dư 10: Cộng A
    - Dư 11: Cộng B
    - Dư 12: Cộng C
    - Dư 13: Cộng D
    - Dư 14: Cộng E
    - Dư 15: Cộng F
'''

def decToBin(number): # hàm chuyển số thập phân -> nhị phân
    bin_number = ''

    if number == 0:
        return "0"
```

```

while number > 0:
    tmp = number % 2
    number = number // 2
    bin_number = bin_number + str(tmp)
return bin_number[::-1]

def decToOct(number): # hàm chuyển số thập phân -> bát phân
    oct_number = ''

    if number == 0:
        return "0"

    while number > 0:
        tmp = number % 8
        number = number // 8
        oct_number = oct_number + str(tmp)
    return oct_number[::-1]

def decToHex(number): # hàm chuyển số thập phân -> thập
lục phân
    hex_digits = "0123456789ABCDEF" # xâu lưu trữ các kí
hiệu hệ Hexa (16)
    hex_number = ""

    if number == 0:
        return "0"

    while number > 0:
        tmp = number % 16
        number = number // 16
        hex_number += hex_digits[tmp] # cộng dồn cho kí tự
tại vị trí tmp trong xâu hex_digits

    return hex_number[::-1]

dec_number = int(input()) # nhập số thập phân

bin_number = decToBin(dec_number) # gán giá trị của các
số chuyển đổi cho các biến tương ứng
oct_number = decToOct(dec_number)
hex_number = decToHex(dec_number)

print(bin_number) # hiển thị kết quả
print(oct_number)
print(hex_number)

```


--

Câu 4: Password weak/strong

Trong các hệ thống CNTT, việc sử dụng mật khẩu là không thể thiếu. Tuy nhiên, việc đặt mật khẩu như nào để đảm bảo mức độ tin cậy thì không phải ai cũng biết. Một mật khẩu mạnh là mật khẩu thỏa mãn các điều kiện sau đây:

- Có tối thiểu 8 kí tự
- Có ít nhất hai chữ cái thường
- Có ít nhất một chữ cái hoa
- Có ít nhất hai chữ số
- Có ít nhất một kí tự đặc biệt: @, #, \$, %, !

Viết chương trình cho phép thực hiện các công việc sau:

- Nhập một chuỗi kí tự từ bàn phím.

Kiểm tra xem đây có phải là mật khẩu mạnh hay không? Nếu là mật khẩu mạnh, hiển thị chữ "Strong Password"; ngược lại hiển thị chữ "Weak Password"

Input: Một chuỗi kí tự

Output: Nếu là mật khẩu mạnh, hiển thị chữ "Strong Password"; ngược lại hiển thị chữ "Weak Password" tương ứng

For example:

Input	Output
Pass@Word123	Strong Password
password	Weak Password

```
#Code:
def check_password_strength(password):    # hàm kiểm tra độ
mạnh của mật khẩu

    # Kiểm tra độ dài của mật khẩu
    if len(password) < 8:
        return "Weak Password"

    # Kiểm tra chữ cái thường
    if not any(char.islower() for char in password):
        return "Weak Password"

    # Kiểm tra chữ cái hoa
    if not any(char.isupper() for char in password):
        return "Weak Password"

    # Kiểm tra chữ số
    if not any(char.isdigit() for char in password):
        return "Weak Password"

    # Kiểm tra kí tự đặc biệt
    special_chars = ['@', '#', '$', '%', '!']
    if not any(char in special_chars for char in password):
        return "Weak Password"

    # Mật khẩu đáp ứng tất cả yêu cầu thuộc các trường hợp
    còn lại
    return "Strong Password"

# Nhập mật khẩu từ bàn phím
password = input()

# Gọi hàm kiểm tra mật khẩu có mạnh không
result = check_password_strength(password)
```

```
print(result)
```

Câu 5: Mã hoá

Như chúng ta đã biết, các phím số trên điện thoại tương ứng với các nhóm ký tự như sau: 2-ABC, 3-DEF, 4-GHI, 5-JKL, 6-MNO, 7-PQRS, 8-TUV, 9-WXYZ. Hãy viết chương trình nhập vào một chuỗi bất kỳ từ bàn phím và mã hóa chuỗi vừa nhập thành các số tương ứng.

- Ví dụ: chuỗi "CNTT-DHSPHN" được mã hóa "2688-347746".

Input: Chuỗi kí tự

Output: Chuỗi kí tự được mã hóa

For example:

Input	Output
CNTT-DHSPHN	2688-347746

```
#Code:
def encode_string(string):    # hàm mã hoá chuỗi
    encoded_string = ''

    for char in string:        # duyệt từng kí tự trong chuỗi
        if char.isalpha():      # nếu là kí tự chữ, thì
            kiểm tra
                char = char.upper()
                if 'A' <= char <= 'C':          # nếu kí tự nằm trong
đoạn từ 'A' đến 'C'
                    encoded_string += '2'
                elif 'D'<= char <= 'F':
                    encoded_string += '3'
                elif 'G' <= char <= 'I':
                    encoded_string += '4'
                elif 'J' <= char <= 'L':
                    encoded_string += '5'
                elif 'M' <= char <= 'O':
                    encoded_string += '6'
                elif 'P' <= char <= 'S':
                    encoded_string += '7'
                elif 'T' <= char <= 'V':
                    encoded_string += '8'
                elif 'W' <= char <= 'Z':
                    encoded_string += '9'
            else:
                encoded_string += char          # ngược lại, nếu là
kí tự số hoặc kí tự đặc biệt, thì giữ nguyên

    return encoded_string

string = input()                # Nhập chuỗi từ bàn phím cần mã
hoá

encoded_string = encode_string(string)    # gọi hàm mã
hoá chuỗi
print(encoded_string)              # hiển thị kết
quả
```

--

Câu 6: Đếm chữ số trong xâu

Hãy viết chương trình thực hiện công việc sau:

1. Cho phép nhập vào một chuỗi các chữ số
2. Thống kê tất cả các chữ số và số lần xuất hiện của chữ số đó trong xâu.

Input: Một chuỗi các chữ số

Output: Các chữ số được sắp xếp theo chiều tăng dần và số lần xuất hiện chữ số đó trong xâu theo ví dụ sau

For example:

Input	Output
1000101010010110	0: 9 1: 7
2312312321232	1: 3 2: 6 3: 4

Giải thích:

Ta thấy bài theo đề cho thì chuỗi đầu vào chỉ gồm các số, vậy nên để thống kê được các số theo thứ tự tăng dần, mình chỉ cần cho vòng lặp chạy từ 0 đến 9. Mà trong chuỗi có hàm count nên nếu chuỗi.count("chuỗi cần tìm") = 0 thì mình không in ra gì cả ngược lại thì mình in ra dưới dạng

```
print(f"chuỗi cần tìm: {chuỗi.count("chuỗi cần tìm")}")
```

Code:

```
n = input()
for i in range(10):
    if (n.count(str(i)) == 0):
        print(f'i: {n.count(str(i))}')
```

Câu 7: Thêm dấu phẩy (ngăn cách hàng nghìn)

Excel có một chức năng cho phép chèn dấu phân tách hàng nghìn cho một số bất kỳ. Hãy viết chương trình mô phỏng lại chức năng này như sau:

1. Cho phép nhập vào một chuỗi các chữ số
2. Thêm dấu phân cách hàng nghìn (dấu phẩy) vào cho số đó

Input: Một chuỗi các chữ số

Output: Chuỗi số được thêm dấu phân cách hàng nghìn (dấu phẩy) theo ví dụ sau

For example:

Input	Output
13324325684	13,324,325,684
842342235.5754	842,342,235.5754

Code:

```
1 #hàm thêm dấu phẩy vào số
2 def format_number(number):
3     # +) Nếu number là số thập phân ta có hai intnum là phần nguyên và
4     # formatNum là phần thực ngược lại formatNum ban đầu rỗng
5     if "." in number:
6         intnum = number[:number.find('.')]
7         formatNum = '.' + number[number.rfind('.')+1:]
8     else:
9         intnum = number
10        formatNum = ''
11    # Đặt biến t là số đếm các chữ số phần nguyên sau khi được xử lý
12    t = 0
13    # for chạy ngược từ cuối xâu intnum lên đầu xâu intnum
14    for i in intnum[::-1]:
15        # với các số ta cộng lần lượt vào đầu formatNum, biến t tăng 1
16        formatNum = i + formatNum
17        t += 1
18        # với t khác độ dài của phần nguyên(intnum) và chia hết cho 3
19        # ta thêm dấu phẩy vào đầu chuỗi formatformatNum
20        if t != len(intnum) and t%3 == 0:
21            formatNum = ',' + formatNum
22    return formatNum
23 number = input()
24 print(format_number(number))
25
```

Câu 8: So sánh GEN

Một tỷ phú người Việt Nam muốn tìm con thông qua việc so sánh DNA. Ông gửi 2 mẫu DNA của mình và của con đến Khoa CNTT nhờ xác định tỷ lệ gen giống nhau. Mỗi DNA được biểu diễn bởi một chuỗi các kí tự, ví dụ: GTCCTG. Để so sánh mức độ tương giống nhau của 2 DNA, người ta sẽ so sánh sự giống nhau của các kí tự ở các vị trí tương ứng. Tỷ lệ giữa số lượng các kí tự giống nhau trên tổng số kí tự của chuỗi DNA chính là tỷ lệ giống nhau của 2 chuỗi DNA. Hãy viết một chương trình hỗ trợ ông tỷ phú với các công việc sau:

- Nhập vào hai chuỗi DNA tương ứng có độ dài bằng nhau.
- Xác định tỷ lệ giống nhau của 2 DNA (độ chính xác lấy 2 chữ số sau dấu chấm thập phân). Nếu hai chuỗi có độ dài khác nhau thì hiển thị chữ "INVALID".

Ví dụ 1: Hai chuỗi DNA nhập vào là: ATTGCCCG và TTTCCTCG

Tỷ lệ giống nhau là $6/8 = 0.75$

Ví dụ 2: Hai chuỗi DNA nhập vào là: ATC và CAT

Tỷ lệ giống nhau là $0/3 = 0$

Code:

```
1 # hàm kiểm tra hai chuỗi dna
2 def checkGen(d1,d2):
3     # xét độ dài của d1 và d2
4     if(len(d1) != len(d2)):
5         return 'INVALID'
6     c = 0 # là biến đếm các kí tự giống nhau ở cùng vị trí d1 d2
7     # đếm các kí tự giống nhau
8     for i in range(len(d1)):
9         if(d1[i] == d2[i]):
10             c+=1
11     # lí do viết trả về như dưới, để có thể dùng trong print() thì kiểu dữ liệu
12     # phải cùng là xâu
13     return f"{c/len(d1):.2f}"
14 d1 = input()
15 d2 = input()
16 print(checkGen(d1,d2))
17
```


Câu 9: Hoán vị tên (Tôn ngộ không)

Trong phim Tây Du Ký, Tôn Ngộ Không bị Hồ lô Từ Kim của Thái Thượng Lão Quân hút khi gọi tên. Mặc dù trả lời với các tên có hoán vị khác nhau như Tôn Hành Giả, Hành Giả Tôn, Hành Tôn Giả, ... hồ lô đều phát hiện ra là 1 người và hút vào trong. Hãy viết một chương trình mô phỏng tác dụng của Hồ lô Từ Kim thực hiện các công việc sau:

- Nhập vào 2 họ tên
- Kiểm tra xem hai họ tên này có phải là hoán vị của nhau hay không. Hiển thị "YES" nếu có, ngược lại là "NO"

Ví dụ 1: Các tên được nhập vào

Ton Hanh Gia
Hanh Gia Ton

Kết quả là : YES

Ví dụ 2: Các tên được nhập vào

Ton Hanh Gia
Hanh Gia Not

Kết quả là: NO

Input:

- Dòng 1: Họ tên thứ nhất
- Dòng 2: Họ tên thứ hai

Ouput: Hiển thị "YES" nếu 2 họ tên là hoán vị của nhau, ngược lại là "NO"

Code:

```
def kiểmtra(str1, str2):  
    tu_str1 = str1.split() #Chia xâu str1 thành các từ  
    tu_str2 = str2.split() #Chia xâu str2 thành các từ  
  
    for tu in tu_str1:  
        if tu not in tu_str2: #Kiểm tra từ trong str1 có trong str2  
            hay không  
            return "NO"  
  
    if len(str1) == len(str2): #Kiểm tra độ dài của hai xâu  
        return "YES"  
    else:  
        return "NO"  
  
a = input()  
b = input()  
print(kiểmtra(a,b))
```

Câu 10: Kiểm tra đăng kiểm

Tại các trung tâm đăng kiểm ô tô của Việt Nam, để xác định việc hết hạn đăng kiểm của một chiếc xe ô tô, người ta sẽ so sánh ngày trên tem đăng kiểm với ngày hiện tại. Hãy viết một chương trình phần mềm hỗ trợ trung tâm đăng kiểm cho phép thực hiện các công việc sau:

- Nhập vào ngày trên tem đăng kiểm và ngày hiện tại (với định dạng dd/mm/yyyy)
- Kiểm tra xem, tem đăng kiểm này hết hạn chưa? Nếu ngày trên tem trước ngày hiện tại thì hiển thị chữ "ĐA HET HAN", ngược lại nếu ngày trên tem sau ngày hiện tại thì hiển thị chữ "CHUA HET HAN". Ngược lại, nếu ngày trên tem đăng kiểm trùng ngày hiện tại thì hiển thị chữ "HOM NAY HET HAN".

Input:

- Dòng 1: ngày trên tem đăng kiểm
- Dòng 2: ngày hôm nay

Output: Hiển thị chữ "CHUA HET HAN", "HOM NAY HET HAN" và "ĐA HET HAN" tương ứng với yêu cầu của đề bài

```
Code:
def split_time(date):
    day = date[:2]
    month = date[3:5]
    year = date[6:]

    return day, month, year

def calculate_time(date_sign, date_now):
    day_sign, month_sign, year_sign = split_time(date_sign)
    day_now, month_now, year_now = split_time(date_now)

    date_sign = year_sign + month_sign + day_sign
    date_now = year_now + month_now + day_now

    if date_sign > date_now:
        print('CHUA HET HAN')
    elif date_sign < date_now:
        print('DA HET HAN')
    else:
        print('HOM NAY HET HAN')

date_sign = input()
date_now = input()

calculate_time(date_sign, date_now)
```

Câu 11: Số đăng kiểm

Tại các trung tâm đăng kiểm ô tô của Việt Nam, để xác định việc hết hạn đăng kiểm của một chiếc xe ô tô, người ta sẽ so sánh ngày trên tem đăng kiểm với ngày hiện tại. Hãy viết một chương trình phần mềm hỗ trợ trung tâm đăng kiểm cho phép thực hiện các công việc sau:

- Nhập vào ngày trên tem đăng kiểm và ngày hiện tại (với định dạng dd/mm/yyyy)
- Kiểm tra xem, tem đăng kiểm này hết hạn chưa? Hiển thị số ngày còn hạn của tem đăng kiểm. Nếu quá hạn thì hiển thị số ngày âm.

Input:

- Dòng 1: ngày trên tem đăng kiểm
- Dòng 2: ngày hôm nay

Output: Số hạn đăng kiểm còn lại của xe

Giải thích:

+) Có thể dễ dàng ta làm được đối với các trường hợp cùng tháng, cùng năm, tuy nhiên với các trường hợp khác năm thì ta có thể làm bằng cách tính tổng số ngày từ năm 0 rồi trừ đi

+) Đầu tiên mình cần tìm năm nhuận (dùng hàm leaf ở dòng 1,2)

cách tìm năm nhuận thì mọi người đã biết ở các bài trước

+) Hàm tính số ngày trong 1 tháng (hàm dom(m,y) với hai tham số là m và y)

-) Tháng 2 năm nhuận: 29

-) Tháng 2 không nhuận: 28

-) Tháng 4,6,9,11: 30

-) Các tháng còn lại: 31

+) Hàm tính số ngày trong các năm (hàm doy(y) với tham số là y):

-) Năm nhuận thì cộng thêm 366

-) Năm không nhuận là 365

+) Hàm tính tổng số ngày (Hàm cal_day(d,m,y) tham số là d,m và y):

-) Bằng ngày hiện tại + số ngày các tháng trước đó + số ngày của năm trước đó

+) Sau khi tính được tổng số ngày của ngày 1 và 2 thì mình lấy hai ngày trừ đi cho nhau như dòng số 23

+) Lưu ý: các tham số ở đây là kiểu số nguyênnguyên

Code:

```
1 def leaf(y):
2     return y%400 == 0 or (y%100 != 0 and y%4 == 0)
3 def dom(m,y):
4     if m == 2:
5         if leaf(y): return 29
6         return 28
7     if m == 4 or m == 6 or m == 9 or m == 11:
8         return 30
9     return 31
10 def doy(y):
11     dy = 0
12     for i in range(y):
13         if(leaf(i)): dy += 366
14         else: dy += 365
15     return dy
16 def cal_day(d,m,y):
17     day = doy(y) + d
18     for i in range(1,m):
19         day += dom(i,y)
20     return day
21 d1,m1,y1 = input().split('/')
22 d2,m2,y2 = input().split('/')
23 print(cal_day(int(d1),int(m1),int(y1)) - cal_day(int(d2),int(m2),int(y2)))
24
```