

## LẬP TRÌNH VỚI PYTHON INT13162 – MD6

### CÂU 1: SLICE STRINGS

Cho input là một chuỗi viết chương trình trả về 9 dòng với yêu cầu như sau:

1. Trong dòng đầu tiên, in độ dài của chuỗi đã cho.
2. Trong dòng thứ hai, in ba ký tự cuối của chuỗi này.
3. Trong dòng thứ ba, bắt đầu bằng ký tự cuối cùng là vị trí đầu tiên 0, in tất cả các ký tự có vị trí là chẵn của chuỗi theo thứ tự ngược từ cuối chuỗi về đầu chuỗi.
4. Trong dòng thứ tư, in tất cả trừ hai ký tự cuối cùng của chuỗi này.
5. Trong dòng thứ năm, in năm ký tự đầu tiên của chuỗi này.
6. Trong dòng thứ sáu, in tất cả các ký tự của chuỗi này với các chỉ số chẵn (hãy nhớ lập chỉ mục bắt đầu từ 0, vì vậy các ký tự được hiển thị bắt đầu bằng chữ cái đầu tiên).
7. Trong dòng thứ bảy, in tất cả các ký tự của chuỗi này với các chỉ số lẻ (tức là bắt đầu bằng ký tự thứ hai trong chuỗi).
8. Trong dòng thứ tám, in tất cả các ký tự của chuỗi theo thứ tự ngược lại.
9. Trong dòng thứ chín, in ký tự thứ hai tính từ cuối lên của chuỗi này.

**Ví dụ:**

Input	Output
qwertyuiop	10 iop piyrw qwertyui qwerty qetuo wryip poiuytrewq o

## CÂU 2: HỆ SỐ JACCARD

Hệ số Jaccard là một đại lượng dùng để đo mức độ tương tự giữa hai chuỗi bất kỳ. Cho hai chuỗi A và B, hệ số Jaccard giữa A và B, được ký hiệu là  $J(A, B)$  và được xác định theo công thức sau:

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|}$$

Cho hai chuỗi A và B, Tính độ đo tương đồng Jaccard giữa hai chuỗi.

Lưu ý:

- Nếu A và B rỗng, thì  $J(A, B)=1$
- $J(A, B)$  nằm trong đoạn  $[0, 1]$
- Lưu ý các kí tự trong chuỗi để tính tương đồng không phân biệt chữ hoa thường, không bao gồm các ký tự như khoảng trắng, dấu chấm (.), dấu phẩy (,) và dấu hỏi chấm (?).

### Input:

Hai chuỗi A và B, mỗi chuỗi được viết trên một dòng.

### Output:

Hệ số Jaccard giữa A và B. Kết quả được in ra với 2 chữ số sau dấu phẩy.

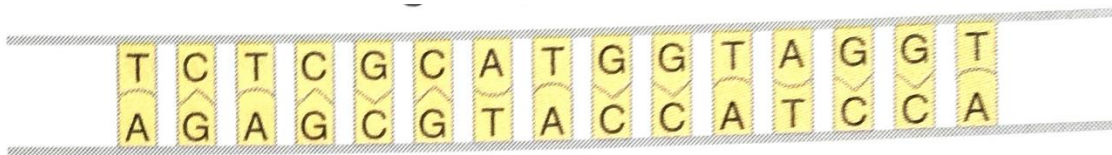
### Ví dụ:

Input	Output
hom nay thi lap trinh python	0.91
Lap trinh Python.	
Hom qua troi mua.	0.50
Hom qua mua, ngay mai co nang khong?	

### CÂU 3: COVID-19 DELTA SUBSTITUTION VARIANT

Biến thể Delta của virus SARS-CoV-2 được phát hiện lần đầu tiên ở Ấn Độ vào tháng 10/2020. Biết rằng, cứ 11 ngày trong quần thể virus xảy ra một đột biến điểm (Nucleotid/ base).

Đây là thuật ngữ của di truyền học phân tử, trong tiếng nước ngoài gọi là **Point Mutation** (tiếng Anh) dùng để chỉ một loại đột biến gen chỉ xảy ra ở một “điểm” trên phân tử di truyền DNA. “Điểm” này tương ứng với một cặp Nucleotide duy nhất.

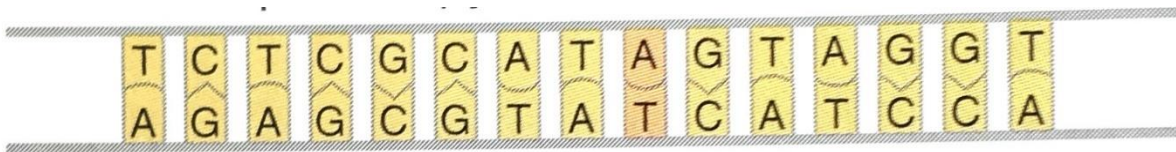


*Hình 1: Bộ gen ban đầu của Virus*

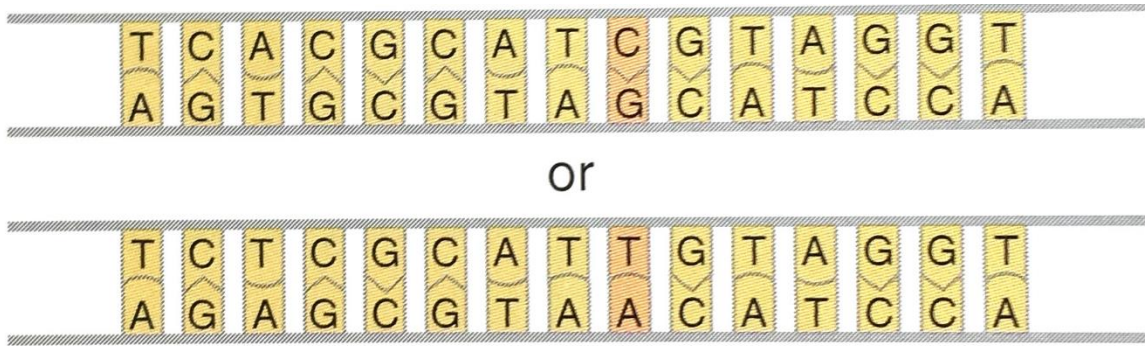
Có ba loại đột biến điểm – Point Mutation là:

- Đột biến thay thế - Substitution
- Đột biến xóa – Deletion
- Đột biến thêm – Insertion

**Đột biến thay thế - Substitution:**



Thay thế một số cặp Nucleotides AT thành GC hoặc thay đổi thứ tự AT thành TA và GC thành CG:



Biết Bộ gen (hay hệ gen) của virus có chiều dài (kích thước) làm tròn là 30.000 Nucleotides / Bases; các đột biến hoàn toàn ngẫu nhiên và với xác suất là như nhau. Giả sử tất cả các đột biến điểm mới của biến thể Delta đều là **đột biến thay thế Substitution** và **thay thế duy nhất 1 cặp**.

**Gợi ý:**

Một đột biến có thể xảy ra tại 1 cặp bất kỳ trên 30.000 cặp Nucleotides nên xác suất 2 đột biến giống nhau là  $1/30.000 \sim 0$  coi như các đột biến sinh ra là khác nhau.

- Sau 11 ngày biến thể Delta ban đầu D0 trên toàn bộ quần thể sẽ tạo ra một đột biến điểm là D01 lúc này ta có 1 **đột biến điểm kiểu Substitution**.
- Sau 11 ngày tiếp theo quần thể Delta D0 sẽ tạo ra thêm 1 biến thể mới là D02 và quần thể D01 sẽ tạo ra thêm 1 đột biến điểm là D012 vậy lúc này ta sẽ có 3 đột biến điểm là D01, D02 và D012....
- .....

Viết chương trình tính số lượng **đột biến điểm kiểu thay thế Substitution** mà biến thể Delta đã có sau ngày 1/10/2020.

**Input:**

Dòng đầu tiên số nguyên  $N$  ( $1 \leq N \leq 10$ ) là số tập test.

$N$  dòng tiếp theo, mỗi dòng gồm một số nguyên có giá trị không vượt quá 365 là số ngày tính từ 1/10/2020 thời điểm biến thể Delta bắt đầu xuất hiện.

**Output**

In ra  $N$  dòng, mỗi dòng một số nguyên là số lượng đột biến điểm của biến thể Delta.

Trường hợp nếu số nguyên  $N$  vượt quá 10 hoặc nhỏ hơn 0 in ra **INVALID INPUT**

Bộ test nào có giá trị lớn hơn 365 in ra dòng đó là **INVALID INPUT**

**Ví dụ**

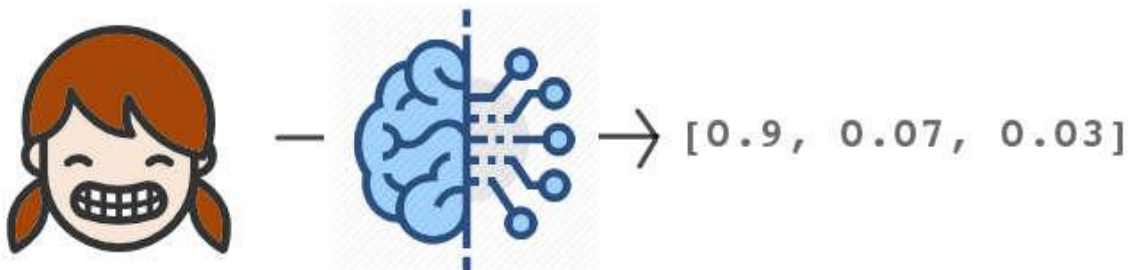
Input:	Output:
5	0
7	INVALID INPUT
380	1
18	1023
113	3
30	

#### CÂU 4: COSINE SIMILARITY

Ứng dụng đo độ tương tự hai vector nhận diện mỗi tấm ảnh đưa vào là khuôn mặt của ai (danh tính) bằng công thức toán tính góc của hai vector:

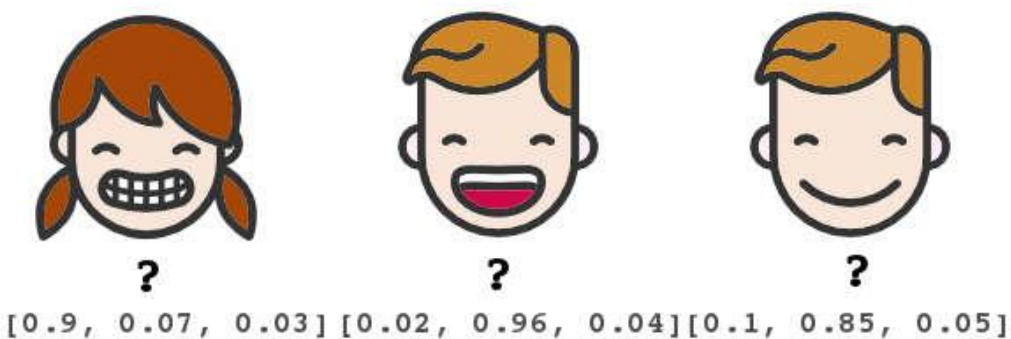
$$similarity = \cos \theta = \frac{\vec{A} \cdot \vec{B}}{|\vec{A}| |\vec{B}|}$$

Trong bài toán xử lý ảnh, đầu ra của giải thuật nhận diện khuôn mặt là 1 vector đặc trưng. Mỗi ảnh trong thư viện ảnh đã có và ảnh cần nhận diện sẽ chạy qua giải thuật trích đặc trưng (**deep learning**) để có vector đặc trưng tương ứng.



Như vậy ta có vector đặc trưng của mỗi khuôn mặt cần nhận diện. Hình minh họa bên dưới:

### Đây là ai?



### Danh tính đã có



Bằng việc so sánh vector đặc trưng của mỗi hình ảnh cần dự đoán **Đây là ai?** với **Danh tính đã có** bằng cách tính độ tương tự giữa hai vector dùng cosine như công thức ở mục phía trên:

### Tính cosine similarity

Bài toán trên đã được số hóa thành:

Ngân =  $[1, 0, 0]$

Minh =  $[0, 1, 0]$

Nhung =  $[0, 0, 1]$

Có một hình ảnh A sau khi được mã hóa là  $A = [0.9, 0.07, 0.03]$

So trùng các vector A với các danh tính đã có Ngân, Minh, Nhung:

$$\cos(A, \text{Ngân}) = \frac{1 \times 0.9 + 0 \times 0.07 + 0 \times 0.03}{\sqrt{1^2 + 0^2 + 0^2} \times \sqrt{0.9^2 + 0.07^2 + 0.03^2}} = 0.9964$$

$$\cos(A, \text{Minh}) = \frac{0 \times 0.9 + 1 \times 0.07 + 0 \times 0.03}{\sqrt{0^2 + 1^2 + 0^2} \times \sqrt{0.9^2 + 0.07^2 + 0.03^2}} = 0.0775$$

$$\cos(A, \text{Nhung}) = \frac{0 \times 0.9 + 0 \times 0.07 + 1 \times 0.03}{\sqrt{0^2 + 0^2 + 1^2} \times \sqrt{0.9^2 + 0.07^2 + 0.03^2}} = 0.0332$$

Kết luận: A là **Ngân** vì có độ tương đồng lớn nhất.

*Viết chương trình tính độ tương đồng cosin của hình ảnh mới với những người trong CSDL và in ra người có độ tương đồng lớn nhất trong hệ thống nhận diện danh tính khi biết vector đặc trưng của mỗi người và một bức ảnh mới A sau khi được mã hóa.*

### Input:

Dòng đầu tiên số nguyên N là tổng số dòng trong bộ test ( $N > 3$ ).

Dòng thứ 2 là số tự nhiên n số chiều của vector đặc trưng cho khuôn mặt ( $n < 1025$ )

Dòng thứ 3 là gồm n số là chỉ số cho từng thành phần của vector đặc trưng cho một bức ảnh mới A

N-3 dòng còn lại bao gồm string Sn đầu tiên là tên người và sau đó là n số tương ứng với n chỉ số cho từng thành phần của vector đặc trưng cho người Sn đó

### Output:

In ra tên người có vector tương đồng lớn nhất với A và in ra danh sách chỉ số độ tương đồng cosin của hình ảnh mới với những người trong CSDL làm tròn đến số thứ 4 sau dấu phẩy.

*Trường hợp số nguyên  $N < 4$  hoặc  $n > 1024$  in ra **INVALID INPUT***

### Ví dụ:

Input:	Output:
6	Ngan [0.9964, 0.0775, 0.0332]
3	
0.9 0.07 0.03	
Ngan 1 0 0	
Minh 0 1 0	
Nhung 0 0 1	

### CÂU 5: TỔNG TÍCH LŨY

Cho một danh sách số gồm N phần tử. Viết chương trình tính tổng tích lũy của một danh sách, nghĩa là, sẽ tạo ra một danh sách mới, trong đó phần tử thứ  $i$  là tổng của  $i+1$  phần tử đầu tiên từ danh sách ban đầu.

Ví dụ: Một danh sách có 3 phần tử: [1, 2, 3] => danh sách tổng tích lũy mới là: [1, 3, 6]

In ra màn hình tích và tổng của các phần tử trong danh sách mới đó.

#### Input:

- Dòng đầu tiên là số phần tử của mảng N.
- Dòng thứ hai là N phần tử của mảng, mỗi phần tử cách nhau bởi khoảng trắng.

#### Output:

- Tổng và tích của danh sách tổng tích lũy, viết trên cùng một dòng.

#### Ví dụ:

Input	Output
3	10 18
1 2 3	



**CÂU 6: DANH SÁCH LỒNG NHAU**

Tên và điểm thi môn Lập trình Python của mỗi sinh viên được lưu vào một danh sách, lưu trữ chúng trong một danh sách lồng nhau. Với một lớp có N ( $N \leq 70$ ) sinh viên, tìm và in ra tên của K ( $K \leq 5$ ) sinh viên có điểm thi cao nhất. Nếu có nhiều sinh viên bằng điểm nhau, hãy sắp xếp tên theo chiều tăng dần của bảng chữ cái.

**Input:**

- Dòng đầu tiên là giá trị N và K, cách nhau bởi một khoảng trắng.
- N dòng tiếp theo là tên và điểm thi của sinh viên

**Output:**

- Tên của K sinh viên có điểm cao nhất, được viết trên một dòng duy nhất mỗi tên cách nhau bởi một khoảng trắng.
- Nếu giá trị N và K không thỏa mãn điều kiện, in ra thông báo: **INVALID INPUT**

**Ví dụ:**

Input	Ouput
10 3	
Hung 6	
Long 7	
Giang 8	
Linh 5	
Tuan 8	Ngan Hoa Khanh
Hoa 9	
Mai 5	
Ngoc 4	
Khanh 9	
Ngan 10	

### CÂU 7: TỪ ĐIỂN

Cho một từ điển gồm K phần tử. Viết chương trình tính tổng và tích của các phần tử có giá trị (value) là số trong từ điển đã cho.

#### Input:

- Dòng đầu tiên là giá trị K ( $K \leq 10$ )
- K dòng tiếp theo là các phần tử của từ điển, mỗi phần tử viết trên một dòng, lần lượt là key và value tương ứng.

#### Output:

- Tổng và tích được viết trên cùng một dòng, cách nhau bởi khoảng trắng.
- Nếu K không thỏa mãn điều kiện thì in ra thông báo **INVALID INPUT**

#### Ví dụ:

Input	Output
4 a 9 b 5 c abc d 1	15 45

## CÂU 8: MONTHLY SALARY

### PHIẾU THU NHẬP THÁNG 10/2021

HỌ VÀ TÊN	Nguyễn Văn A	EMAIL	Nva@abc.com.vn
Chức danh	DEV	Bộ phận	IT

NGÀY CÔNG CHUẨN	22	CÁC KHOẢN KHẤU TRỪ (2)	000.000
Ngày công chính thức	22	BHXX (8%*LCB)	000.000
Ngày công thử việc	0	BHYT (1,5%*LCB)	00.000
<b>THU NHẬP THỰC TẾ (1)</b>	<b>00.000.000</b>	BHTN (1%*LCB)	00.000
Lương cơ bản chính thức (LCB)	0.000.000	Đoàn phí (1%LCB)	00.000
Thưởng đánh giá tháng chính thức (LT)	00.000.000	Thuế TNCN	00.000
Lương cơ bản thử việc	0		
Thưởng đánh giá tháng thử việc	0		
Trợ cấp 1 (Không chịu thuế TNCN)			
Trợ cấp 2 (Không chịu thuế TNCN)			
<b>THỰC LĨNH (3) = (1) - (2)</b>	<b>00.000.000</b>		

Trên đây là Phiếu thu nhập của một người trong một tháng bao gồm:

- Thu nhập thực tế (1)
- Các khoản khấu trừ (2)
- Thực lĩnh (3)

Thu nhập thực tế (1) = Lương cơ bản (LCB) + Lương thưởng đánh giá tháng (LT)

Các khoản khấu trừ (2) = BHXX + BHYT + BHTN + ĐP + TNCN

Thực lĩnh (3) = Thu nhập thực tế (1) - Các khoản khấu trừ (2)

BHXX (Bảo hiểm xã hội) = 8%\*LCB

BHYT (Bảo hiểm y tế) = 1,5%\*LCB

BHTN (Bảo hiểm thất nghiệp) = 1%\*LCB

ĐP (Đoàn phí) = 1%\*LCB

Thuế TNCN (Thuế Thu nhập cá nhân) = Thu nhập chịu thuế (TNTT)\*Thuế suất

Thu nhập chịu thuế được tính bằng:

$TNTT = \text{Thu nhập thực tế (1)} - \text{Các khoản giảm trừ} - \text{Tiền đóng bảo hiểm}$

Tiền đóng bảo hiểm (BH) = BHXH + BHYT + BHTN

Các khoản giảm trừ = Giảm trừ bản thân (11.000.000 VNĐ) + Người phụ thuộc (4.400.000VNĐ / 1 người)

Dựa vào điều kiện của Thu nhập chịu thuế (TNTT) như bảng dưới đây ta sẽ có cách tính Thuế TNCN:

Bậc	Thu nhập tính thuế /tháng	Thuế suất	Tính số thuế phải nộp	
			Cách 1	Cách 2
1	Đến 05 trđ	5%	0 trđ + 5% TNTT	5% TNTT
2	Trên 05 trđ đến 10 trđ	10%	0,25 trđ + 10% TNTT trên 5 trđ	10% TNTT - 0,25 trđ
3	Trên 10 trđ đến 18 trđ	15%	0,75 trđ + 15% TNTT trên 10 trđ	15% TNTT - 0,75 trđ
4	Trên 18 trđ đến 32 trđ	20%	1,95 trđ + 20% TNTT trên 18 trđ	20% TNTT - 1,65 trđ
5	Trên 32 trđ đến 52 trđ	25%	4,75 trđ + 25% TNTT trên 32 trđ	25% TNTT - 3,25 trđ
6	Trên 52 trđ đến 80 trđ	30%	9,75 trđ + 30% TNTT trên 52 trđ	30 % TNTT - 5,85 trđ
7	Trên 80 trđ	35%	18,15 trđ + 35% TNTT trên 80 trđ	35% TNTT - 9,85 trđ

Ví dụ: TNTT = 6.000.000VNĐ thì thuế TNCN = 250.000 + 10\*(TNTT – 5.000.000) = 350.000VNĐ

Viết chương trình in ra số tiền Thực lĩnh của người lao động trong 1 tháng khi biết Thu nhập thực tế (1) và LCB của người đó hàng tháng và người này không có người phụ thuộc

### Input:

Dòng đầu tiên số nguyên N ( $1 \leq N \leq 10$ ) là số tập test.

N dòng tiếp theo, mỗi dòng gồm 2 số m, n lớn > 0. Trong đó m là Thu nhập thực tế (1) và n là LCB

### Output:

In ra 1 số là tiền Thực lĩnh hàng tháng (lấy đến phần nguyên)

*Trường hợp số nguyên N vượt quá 10 hoặc nhỏ hơn 0 in ra **INVALID INPUT***

*Bộ test nào có giá trị nhỏ hơn 1000 in ra dòng đó là **INVALID INPUT***

**Ví dụ:**

Input:	Output:
4	14773575
15700000 6300000	12559425
13300000 5700000	78097475
100000000 7700000	INVALID INPUT
1000 123	

### **CÂU 9: KHOẢNG CÁCH HAVERSINE**

Khoảng cách Haversine là một đại lượng để xác định khoảng cách địa lý giữa hai địa điểm bất kỳ. Giả sử hai địa điểm  $P_1$  và  $P_2$  được quy đổi về kinh độ và vĩ độ tương ứng là  $long_1, lat_1, long_2, lat_2$ . Khoảng cách Haversine  $d$  được xác định theo công thức sau:

$$a = \sin^2\left(\frac{\Delta lat}{2}\right) + \cos(lat_1) \cdot \cos(lat_2) \cdot \sin^2\left(\frac{\Delta long}{2}\right)$$

$$c = 2 \cdot \arcsin(\sqrt{a})$$

$$d = R \cdot c$$

Trong đó:

$$\Delta lat = lat_2 - lat_1$$

$$\Delta long = long_2 - long_1$$

$R = 6731$  (km) (Bán kính Trái Đất)

Hãy viết hàm tính khoảng cách Haversine giữa hai địa điểm bất kỳ.

Input:

- Kinh độ và vĩ độ tương ứng của hai địa điểm, mỗi địa điểm được viết trên một dòng, mỗi giá trị cách nhau bởi khoảng trắng và là các số thực (float).

Output:

- Khoảng cách Haversine giữa hai địa điểm, kết quả được in ra với hai số sau dấu phẩy.

Ví dụ:

Input	Output
105.96 10.21	4686.47
107.17 16.79	

### CÂU 10: TUYỂN NHÂN VIÊN

Doanh nghiệp X cần tuyển một số nhân viên mới. Bài thi tuyển có hai phần: lý thuyết và thực hành. Sau khi tính điểm trung bình, các thí sinh sẽ được xếp thành 4 loại:

- Nếu điểm dưới 5 -> TRUOT
- Nếu điểm lớn hơn hoặc bằng 5 nhưng nhỏ hơn 8 -> CAN NHAC
- Nếu điểm từ 8 đến 9.5 -> DAT
- Nếu điểm lớn hơn 9.5 -> XUAT SAC

Điểm các bài thi lý thuyết và thực hành đều là số thực trong phạm vi từ 0 đến 10. Tuy nhiên, khi nhập điểm các bài thi, cán bộ tuyển dụng có thể quên mất dấu . phân cách phần nguyên và phần thập phân. Do đó nếu điểm ghi là 78 thì cần được hiểu là 7.8

Hãy sắp xếp danh sách thí sinh đã được xếp loại theo điểm trung bình giảm dần. Nếu có nhiều thí sinh bằng điểm nhau, hãy sắp xếp tên theo chiều tăng dần trên bảng chữ cái.

### Input

Dòng đầu ghi số thí sinh. Mỗi thí sinh ghi trên 3 dòng lần lượt là:

- Họ và tên (xâu ký tự độ dài không quá 100)
- Điểm lý thuyết

- Điểm thực hành

## Output

Ghi ra danh sách thí sinh đã sắp xếp, mỗi thí sinh gồm 3 thông tin: họ tên, điểm trung bình (với 2 số phần thập phân) và xếp loại. Mỗi thông tin cách nhau một khoảng trống.

Input	Output
3	
Nguyen Thai Binh	
45	
75	
Le Cong Hoa	Nguyen Thai Binh 6.00 CAN NHAC
4	Phan Van Duc 5.60 CAN NHAC
4.5	Le Cong Hoa 4.25 TRUOT
Phan Van Duc	
56	
56	