

### Câu 1:

#### Tích Lẻ chia Tổng Chẵn

Viết chương trình tính giá trị của một dãy số bằng cách lấy tích các số tại vị trí lẻ (bỏ qua giá trị 0) chia cho tổng giá trị các số tại vị trí chẵn. Nếu tổng các số tại vị trí chẵn bằng 0 thì in ra INVALID

Input

- Dòng đầu ghi số bộ test
- Các dòng tiếp theo, mỗi dòng ghi một số có số chữ số không quá 100

Ouput

- Giá trị tính được theo yêu cầu, làm tròn đến 6 số sau dấu phẩy. Nếu tổng bằng 0 thì in ra INVALID

Ví dụ:

Input	Output
2	0.500000
1012	INVALID
30304	INVALID

Giải thích test

test1: tích lẻ =  $1 \times 1 = 1$ , tổng chẵn:  $0 + 2 = 2$ , kết quả là 0.5

test2: tích lẻ =  $3 \times 3 \times 4 = 36$ , tổng chẵn:  $0 + 0 = 0$ , kết quả là INVALID

### Câu 2:

#### Độ tương tự Jaccard

Độ tương tự Jaccard giữa 2 vector trong toán học được định nghĩa với công thức:

$$\text{Jaccard} = \frac{\text{intersection}(A, B)}{\text{union}(A, B)}$$

trong đó A và B là hai vector n chiều. Tử số là số các phần tử chung của A và B, mẫu số là số các phần tử xuất hiện ở A hoặc xuất hiện ở B

Viết chương trình tính Độ tương tự Jaccard của hai vector A và B.

Input

- Dòng đầu ghi số bộ test
- Hai dòng tiếp theo ghi 2 vector, mỗi vector là một bộ số trong đó các số cách nhau bởi một hoặc nhiều dấu cách

Ouput:

- Độ tương tự của 2 vector, làm tròn đến 5 số sau dấu phẩy. Nếu 2 vector không cùng số chiều thì in ra INVALID

Ví dụ

Input	Output
2	
1 4 3 2	
0 0 1 2	0.40000
2 4 6	
2 0	

#### Câu 3:

##### Tạo từ điển các từ khoá

Viết một hàm nhận một dictionary và thực hiện các công việc sau:

- Trả về danh sách các **key** có giá trị là số chẵn.
- Tính tổng tất cả các giá trị là số.
- Đếm số lượng key có giá trị là chuỗi.

Hàm phải trả về một tuple gồm 3 phần:

- Danh sách các key có giá trị là số chẵn.
- Tổng tất cả các giá trị là số.
- Số lượng key có giá trị là chuỗi.

#### Input

Dòng đầu tiên ghi số bộ test.

Mỗi bộ test gồm một dòng chứa một từ điển *A*

#### Output

Với mỗi bộ test, in ra Danh sách các key có giá trị là số chẵn, Tổng tất cả các giá trị là số. Và số lượng key có giá trị là chuỗi..

#### Ví dụ

Input	Output
5	
{"a": 2, "b": 3, "c": "hello", "d": 4}	(['a', 'd'], 9, 1)
{"x": 10, "y": 20, "z": "world"}	(['x', 'y'], 30, 1)
{"key1": 1, "key2": 5, "key3": "python"}	([], 6, 1)
{"p": "test", "q": "code", "r": 0}	(['r'], 0, 2)
{}	([], 0, 0)

#### Câu 4:

##### Chấm công

Công ty chấm công các nhân viên để tính thời gian làm việc, từ đó tính xem nhân viên có làm việc đủ thời gian không. Thời gian làm việc chuẩn là 8 giờ. Thời gian làm việc của công nhân tính bằng thời gian ra về trừ thời gian đến và trừ một giờ (60 phút) nghỉ trưa.

#### Input

Dòng đầu của dữ liệu vào ghi số lượng nhân viên chấm công (không quá 20).

Thông tin về một công nhân chấm công được ghi lại trên 4 dòng lần lượt là:

- Mã nhân viên (xâu ký tự độ dài không quá 10, không có khoảng trắng)
- Tên người chơi (xâu ký tự độ dài không quá 100, có thể có khoảng trắng).
- Giờ vào (định dạng hh:mm)
- Giờ ra (định dạng hh:mm)

### Ouput

Ghi ra danh sách nhân viên đã được sắp xếp theo thời gian làm việc giảm dần cùng với thông tin thiếu (THIEU) hoặc đủ thời gian tiêu chuẩn (DU).

### Ví dụ

Input	Output
2 01T Nguyen Van An 08:00 17:30 06T Tran Hoa Binh 09:05 17:00	01T Nguyen Van An 8 gio 30 phut DU 06T Tran Hoa Binh 6 gio 55 phut THIEU

### Câu 5:

#### **GEOLOGY PROBLEMS**

Cho  $N$  điểm trên mặt phẳng Oxy. Nhiệm vụ của bạn là xác định xem có tồn tại một đường tròn ngoại tiếp của 3 đỉnh và thỏa mãn có đúng  $K$  điểm **nằm trong** đường tròn hay không (không tính các điểm nằm trên đường tròn).

#### **Dữ liệu vào:**

Dòng đầu tiên là số lượng bộ test  $T$  ( $T \leq 10$ ).

Mỗi test gồm số nguyên  $N$  ( $3 \leq N \leq 100$ ), là số lượng điểm trên mặt phẳng. Dòng tiếp theo là số nguyên  $K$  ( $1 \leq K \leq 100$ ).

$N$  dòng tiếp theo, dòng thứ  $i$  gồm 2 số nguyên  $x_i, y_i$  ( $-1000 \leq x_i, y_i \leq 1000$ ).

#### **Kết quả:**

Với mỗi test, in ra "YES" nếu tìm được đường tròn chứa đúng  $K$  điểm. In ra "NO" trong trường hợp ngược lại.

#### **Ví dụ:**

#### **Input Output**

```
2      YES
4      NO
1
```

0 0  
5 0  
0 5  
1 1  
5  
2  
5 5  
5 -5  
-5 5  
-5 -5  
0 0