Question 1: Normal Parking Slot Availability

A parking lot is represented as a **2D array**, where:

- 1 → filled (occupied slot)
- 0 → free (available slot)

Task:
Print all the available parking slots in the format:

Slot available at X-Row and Y-Column

Example Input

Parking Lot (2D Array):

1 0 1 0

0 1 0 1

1 1 0 0

Example Output

Slot available at 1st Row and 2nd Column

Slot available at 1st Row and 4th Column

Slot available at 2nd Row and 1st Column

Slot available at 2nd Row and 3rd Column

Slot available at 3rd Row and 3rd Column

Slot available at 3rd Row and 4th Column

Question 2: Contiguous Parking Slot Availability

The same parking lot is represented as a **2D array**, where:

- 1 → filled (occupied slot)
- 0 → free (available slot)

Task:
Check if there are **k contiguous free slots** in any row.
If found, print in the format:

k-contiguous slots available at X-Row starting from Y-Column

If not found, print:

No k-contiguous slots available

Example Input

Parking Lot (2D Array):

1 0 0 0 1

1 1 0 0 0

0 1 1 0 0

k = 3

Example Output

3-contiguous slots available at 1st Row starting from 2nd Column

3-contiguous slots available at 2nd Row starting from 3rd Column

Coding Question: Equalizing Water Tanks

Description:

Two water tanks contain **X** liters and **Y** liters of water. You are given a list of **N available water containers** (each container has a fixed capacity).

You can pour water from **at most two containers** into either of the tanks.
Your task is to determine whether it is possible to make the two tanks have **equal water levels**.

Input Format

- First line will contain an integer T, denoting the number of test cases.
  For each test case:

1. First line will contain two positive integers denoting X and Y (liters of water in the tanks).
2. Second line will contain an integer N, the number of available containers.
3. Third line will contain N space-separated integers, denoting the container capacities.

Output Format

For each test case, print YES if the tanks can be equalized, else print NO, in new lines.

Constraints

1 <= T <= 10

1 <= X, Y <= 10^6

1 <= N <= 10^5

1 <= container[i] <= 10^5

Sample Input

3

3 7

5

2 3 5 18 4

5 8

5

2 9 4 11 6

4 8

5

2 1 1 9 7

Sample Output

YES

YES

NO

Explanation

- **Case 1**: X = 3, Y = 7 → difference = 4 → container with 4 can be added to smaller tank → Balanced ✅
- **Case 2**: X = 5, Y = 8 → difference = 3 → no single container is 3, but 9 added to 5 and 6 added to 8 makes both tanks = 14 ✅
- **Case 3**: X = 4, Y = 8 → difference = 4 → but no 4 available, and no two containers can fix it ❌