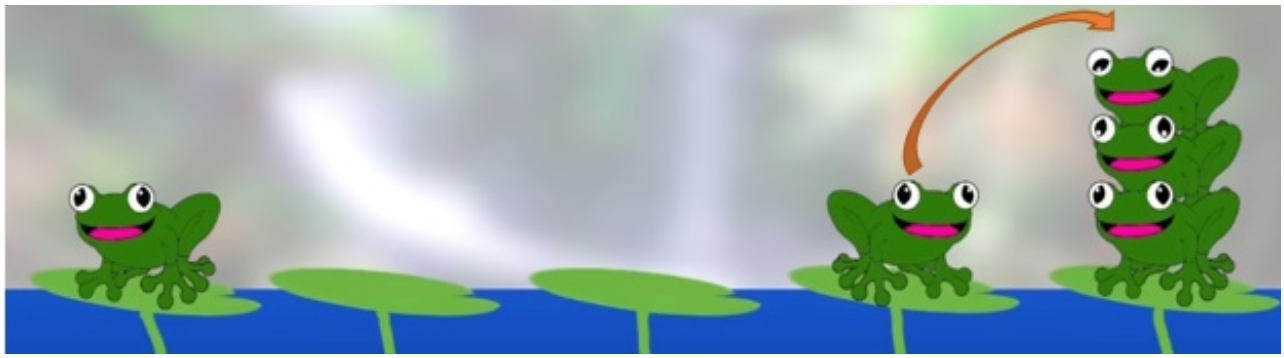
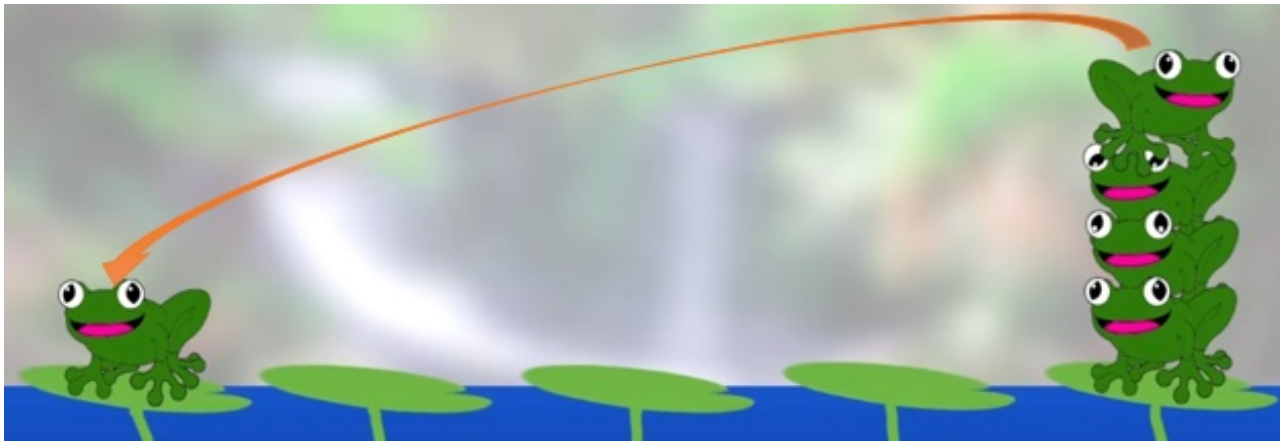
Frog Party

In the local pond there are **n** lillypads arranged in a row - frog households - numbered **1** to **n** from left to right. One frog lives on each lillypad. In the afternoon, each frog returns to its household for a short nap. After they wake up from their naps, it is already evening. And what do frogs do on evenings? They throw a party! Luckily, a frog party doesn't require any special preparations - all the frogs simply meet up at a single lillypad, and the fun may begin.

When a frog goes to a party, it definitely doesn't want to arrive wet. Hence, frogs may only travel to the party by jumping across lillypads. A single frog by itself can jump to a neighbouring lillypad. However, when a frog jumps to a lillypad with another frog on it, they team up - the frog that arrived jumps on the other's back, and with combined efforts (and slightly defying the laws of physics) they can jump to a lillypad which is at a distance of **2**. In general, if there are **y** frogs on the lillypad numbered **x**, they can jump to lillypads with numbers **x-y** and **x+y** (jumping outside the **n** lillypads is obviously forbidden - the frogs would jump into the water and get wet). When frogs from one lillypad jump onto another, they climb on the backs of the frogs which were already present, in the same order as before. On top of that, polite frog manners say that frogs should not jump on a lillypad on which there are currently no frogs.

The whole process might look like this





As you can imagine, when several frogs stand on another's back, it can be quite tiring. Frog Michal obviously doesn't want to do activities which are tiring. That's why he would like to propose a sequence of jumps at the next frog meeting, such that in the evening all frogs will end up meeting on one lillypad, and at the same time no frog ever climbed on Michal's back (that basically means that no frog can ever jump to the lillypad on which Michal is currently situated). Help him!

Given the number of lillypads **n** and the number of lillypad on which Michal lives, produce a sequence of jumps so that all frogs end up on a single lillypad, and no frog ever jumps on a lillypad which currently has no frogs on it, or has frog Michal on it.

Input

The first line contains an integer **1 ≤ T ≤ 60**: the number of testcases.

For each testcase, in the only line of input there are two whole numbers **n** and **m**, **1 ≤ m ≤ n ≤ 10^6**: the number of lillypads (and frogs), and the number of the lillypad on which Michal lives.

The sum of **n** within a single input file does not exceed **$2*10^6$**.

Output

For each testcase, if no valid sequence of jumps which fulfills the criteria described exists, output "NO". Otherwise in the first line output "YES", and then describe a valid sequence of jumps. Each jump can be described by two numbers **1 ≤ a,b ≤ n**, meaning that the frogs currently on lillypad **a** should jump to the lillypad **b**. Any valid sequence of jumps leading to a frog party will be accepted.

The output is very large. Please be wary of your I/O speed.

Example

Input:

```
2
5 4
9 3
```

Output:

```
YES
2 3
3 5
4 5
5 1
YES
1 2
3 2
4 5
6 5
7 8
9 8
8 5
2 5
```

The first case is depicted in the images in the problem text.