

Very Friends 2

You are creating a new social network for dogs. Wow. The dogs don't have many possibilities for interacting with your website, but they can bark how many friends they want. E.g. if a dog wants to have much 8 friends it will bark 8 times, and if it doesn't want any friends, it'll just stay quiet.

After spending a good year of your life collecting these barks, you are finally ready to assign a friend list for each dog. The only problem is: You are not sure whether it is actually possible. Thus before you proceed you would like to write a program, that given a list of **N** wishes w_i , outputs **HAPPY** if it is possible to make a friend list for each dog **i** of length w_i , or **SAD** if some dog will have to get more or fewer friends than it wished for.

Notice: Being friends is considered a reflexive relation.

Input

The first line will contain a single integer **T** - the number of test cases to process.

Because of I/O constraints, the sequence of wishes is not given explicitly. Each of the **T** lines will consist of 5 integers **N**, **a**, **b**, **c**, **m** in the range **[0, 10⁷]** (except **m** which is in **[1, 10⁷]**). These integers describe the sequence

$$x_0 = 0$$

$$x_{i+1} = (a \cdot x_i + b) \% m$$

The sequence of wishes is $w_i = x_i + c$.

Output

Write the answer - **HAPPY** or **SAD** - for each test case on a separate line.

Example

Input:

```
3
3 2 1 0 2
5 1 1 0 5
6 1 1 1 3
```

Output:

```
HAPPY
SAD
HAPPY
```

Explanation

In the first case we get the wishes "0 1 1", and we can make dog 2 and 3 be friends.

In the second case we get the wishes "0 1 2 3 4". No assignment that works, since dog 5 would have to be friends with everyone, but dog 1 doesn't want that.

In the third case we get the wishes "1 2 3 1 2 3".