Unbreakable Diamonds

After finding out that his starship failed some safety tests due to the fragility of the materials he used in building it, Jojo has now decided to use diamonds to build his starship. Jojo then goes to see his friend Bibi, a diamond specialist, to ask for her advice.

Bibi then told Jojo that every diamond has a class assigned to it, represented by a string S. For each diamond, Jojo can find out whether it is unbreakable by checking the amount of distinct characters in its class name. A diamond is unbreakable if and only if the amount of distinct characters in its class name is odd.

Since there are hundreds of diamonds and Jojo has no time to check them all, Jojo has decided to ask for your help again. Help Jojo by creating a program that tells Jojo whether each diamond is unbreakable!

Format Input

The input consists of T test cases, representing the number of diamonds Jojo needs to check.

Each test case consists of a single string S containing only lowercase letters, representing the class name of each diamond.

Format Output

For each test case, output one line containing "Case #X: " (without quotes), where X is the test case number (starting from 1), then followed by either "Breakable" (without quotes) or "Unbreakable" (without quotes), depending on whether the diamond is breakable or not.

Constraints

- $1 \leq T \leq 100$
- $1 \leq |S| \leq 10^5$

$|S|$ means the length of string S.

Sample Input 1 (standard input)

```
1
abb
```

Sample Output 1 (standard output)

```
Case #1: Breakable
```

Sample Input 2 (standard input)

```
2
abbc
za
```

Sample Output 2 (standard output)

```
Case #1: Unbreakable
Case #2: Breakable
```

Note

In the first sample input, the diamond is breakable because there are 2 different letters in its class name, namely “a” and “b”, so the correct output is “Breakable”.

Unbreakable Diamonds

Setelah Jojo tahu bahwa starship yang ia rakit gagal melewati beberapa tes keamanan karena material yang ia gunakan kurang kuat, Jojo memutuskan untuk menggunakan berlian untuk merakit starship miliknya. Jojo kemudian pergi ke temannya Bibi, seorang ahli berlian, untuk meminta sarannya.

Bibi kemudian memberi tahu Jojo bahwa setiap berlian memiliki sebuah kelas yang telah ditentukan, direpresentasikan dengan sebuah string S. Untuk setiap berlian, Jojo dapat mencari tahu apakah berlian tersebut dapat dihancurkan atau tidak dengan memeriksa jumlah karakter berbeda yang ada di nama kelas tersebut. Sebuah berlian tidak dapat dihancurkan jika dan hanya jika jumlah karakter berbeda yang ada di nama kelasnya ganjil.

Karena ada ratusan berlian yang harus diperiksa dan Jojo tidak memiliki cukup waktu untuk memeriksa semuanya, Jojo memutuskan untuk kembali meminta bantuanmu. Bantulah Jojo dengan membuat sebuah program yang memberi tahu Jojo apakah suatu berlian dapat dihancurkan.

Format Input

Input terdiri dari T test case (kasus uji), merepresentasikan jumlah berlian yang harus Jojo periksa.

Setiap test case terdiri dari sebuah string S yang hanya terdiri dari huruf kecil, merepresentasikan nama kelas dari tiap berlian.

Format Output

Untuk setiap test case, tampilkan sebuah baris berisi "Case #X: " (tanpa kutip), dimana X merupakan nomor test case (dimulai dari 1), kemudian diikuti oleh "Breakable" (tanpa kutip) jika berlian tersebut dapat dihancurkan atau "Unbreakable" (tanpa kutip) sebaliknya.

Constraints

- $1 \leq T \leq 100$
- $1 \leq |S| \leq 10^5$

$|S|$ means the length of string S.

Sample Input 1 (standard input)

```
1
abb
```

Sample Output 1 (standard output)

```
Case #1: Breakable
```

Sample Input 2 (standard input)

```
2
abbc
za
```

Sample Output 2 (standard output)

```
Case #1: Unbreakable
Case #2: Breakable
```

Note

Di sample input pertama, berlian tersebut dapat dihancurkan karena ada 2 huruf berbeda di nama kelasnya, yaitu “a” dan “b”, sehingga output yang benar adalah “Breakable”.