**E For Encryption**

**Deskripsi Soal:**

Enkripsi adalah metode mengubah data menjadi kode yang sulit dipahami. Dan kamu memiliki teman bernama whoami yang suka sekali bermain dengan enkripsi. Hingga pada suatu hari, whoami membuat enkripsi nya sendiri. Enkripsi whoami memiliki aturannya sendiri. Berikut adalah aturan dari enkripsi whoami:

* Setiap karakter akan dienkripsi dengan menggunakan aturan yang ada.
* Karakter alphabet akan diubah menjadi angka sesuai dengan urutan, dimana ‘A’ akan berubah menjadi 01, seterusnya sampai ‘Z’ berubah menjadi 26. Dan aturan ini tidak case sensitive.
* Karakter numerik tidak akan ada yang berubah, dan tetap sama nilainya.
* Karakter non-alphabet-numerik akan diubah menjadi karakter yang diulang sebanyak 3x.
* Akan terdapat spasi pemisah pada hasil enkripsi, apabila karakter sebelum dienkripsi adalah karakter numerik, dan karakter non-alphabet-numerik.
* Akan terdapat spasi pemisah pada hasil enkripsi, apabila hasil penjumlahan masing masing angka dari karakter alphabet setelah di enkripsi dengan karakter alphabet selanjutnya setelah dienkripsi tidak sama.

Bantulah whoami untuk membuat sebuah program berdasarkan aturan enkripsi whoami.

**Notes:**

* **Maksud dari non-alphabet-numerik adalah karakter seperti ./@#%&:,**
* **‘**hasil penjumlahan masing masing angka dari karakter alphabet setelah di enkripsi’.

**Contoh maksud dari kata tersebut:**

**A -> 01 -> 1 -> Hasil jumlah nya adalah 1**

**Z -> 26 -> 8 -> Hasil jumlah nya adalah 8**

**Format Input:**

Input baris pertama berisi T, dimana T adalah jumlah banyak testcase. Untuk setiap testcase, terdapat string S tanpa spasi yang akan diubah menggunakan aturan enkripsi whoami.

**Format Output:**

Untuk setiap test case, hasilkan “Case #X: E”, di mana X adalah nomor testcase mulai dari 1, dan E merupakan hasil dari yang sudah di enkripsi.

**Constraints:**

1 <= T <= 10

1 <= Panjang String S <= 100

**Sample Input 1 (Standard Input):**

3

HelloB28,GoodLuckOnYourCompetition.

ThisProblemIsEasy,ButTricky.

HaveFunWithIt:)~NS

**Sample Output 1 (Standard Output):**

Case #1: 08 05 1212 15 02 02 08 ,,, 07 1515 04 122103 11 15 14 25 15 21 18 03 15 13 16 05 20 09 20 09 15 14 ...

Case #2: 20 08 09 19 16 18 15 02 12 05 13 09 19 05 01 19 25 ,,, 02 21 2020 1809 03 11 25 ...

Case #3: 08 01 22 05 06 21 1423 09 20 08 09 20 ::: ))) ~~~ 14 19

**Sample Input 2 (Standard Input):**

1

WelcomeB28

**Sample Output 2 (Standard Output):**

Case #1: 2305 1203 15 13 05 02 02 08

**Penjelasan Case:**

Terdapat satu input String, yaitu “WelcomeB28”. Dan hasil convert nya adalah sebagai berikut:

* W -> 23 -> 5
* e -> 05 -> 5
* l -> 12 -> 3
* c -> 03 ->. 3
* o -> 15 -> 6
* m -> 13 -> 4
* e -> 05 -> 5
* B -> 02 -> 2
* 2 -> 02 -> 2
* 8 -> 08 -> 8

Perhatikan hasil enkripsi ‘W’ dan ‘e’, jumlah hasil nilai enkripsinya sama, sehingga pada hasil akhir output, tidak ada spasi pemisah. Dan ini juga berlaku pada karakter ‘l’ dan ‘c’. Meskipun karakter ‘B’ dan ‘2’ memiliki jumlah hasil yang sama, tapi karakter ‘2’ merupakan karakter numerik, sehingga terdapat spasi pemisah.

*(Jangan lupa sertakan enter ‘\n’ pada setiap output)*

**E For Encryption**

**Case Description:**

Encryption is a method of turning data into a code that is hard to understand. And you have a friend named whoami who loves playing with encryption. Until one day, whoami makes his own encryption. Whoami encryption has its own rules. Here are the rules of whoami encryption:

* Each character will be encrypted using existing rules.
* Alphabetic characters will be converted into numbers in the order in which 'A' will turn into 01, and so on until 'Z' turns into 26. And this rule is not case sensitive.
* Numeric characters will not be changed, and will remain the same value.
* Non-alphabetic-numeric characters will be converted into characters that are repeated 3x.
* There will be separating spaces in the encryption result, if the characters before encryption are numeric characters, and non-alphabet-numeric characters.
* There will be separating spaces in the encryption result, if the sum of each number of alphabetic characters after encryption and the next alphabetic character after encryption is not the same.

Help whoami to create a program based on whoami’s encryption rule

**Notes:**

* **By non-alphabetic-numeric, it means characters like ./@#%&:,**
* 'the sum of each number of alphabetic characters after encryption'.

**Example of what this means:**

**A -> 01 -> 1 -> The sum result is 1**

**Z -> 26 -> 8 -> The sum result is 8**

**Input Format:**

The first line of input contains T, where T is the number of testcases. For each testcase, there is a string S without spaces that will be transformed using whoami encryption rules.

**Output Format:**

For each test case, produce “Case #X: E”, where X is the testcase number starting from 1, and E is the encrypted result.

**Constraints:**

1 <= T <= 10

1 <= Length of String S <= 100

**Sample Input 1 (Standard Input):**

3

HelloB28,GoodLuckOnYourCompetition.

ThisProblemIsEasy,ButTricky.

HaveFunWithIt:)~FromNS

**Sample Output 1 (Standard Output):**

Case #1: 08 05 1212 15 02 02 08 ,,, 07 1515 04 122103 11 15 14 25 15 21 18 03 15 13 16 05 20 09 20 09 15 14 ...

Case #2: 20 08 09 19 16 18 15 02 12 05 13 09 19 05 01 19 25 ,,, 02 21 2020 1809 03 11 25 ...

Case #3: 08 01 22 05 06 21 1423 09 20 08 09 20 ::: ))) ~~~ 06 18 15 13 14 19

**Sample Input 2 (Standard Input):**

1

WelcomeB28

**Sample Output 2 (Standard Output):**

Case #1: 2305 1203 15 13 05 02 02 08

**Case Explanation:**

There is one String input, which is “WelcomeB28”. And the conversion result is as follows:

* W -> 23 -> 5
* e -> 05 -> 5
* l -> 12 -> 3
* c -> 03 ->. 3
* o -> 15 -> 6
* m -> 13 -> 4
* e -> 05 -> 5
* B -> 02 -> 2
* 2 -> 02 -> 2
* 8 -> 08 -> 8

Notice the encryption results of 'W' and 'e', the sum of the encryption value results is the same, so in the final output result, there is no separator space. And this also applies to the characters 'l' and 'c'. Although the characters 'B' and '2' have the same number of results, the character '2' is a numeric character, so there is a separating space.

(Don't forget to include enter '\n' in each output)