

IEEEExtreme Türkiye Kampı: Gün 4 SRAINFUCK (Simple Brainfuck)

Problem

Brainfuck turing-bütün(turing-complete) olan ezoterik bir dildir.

(<https://en.wikipedia.org/wiki/Brainfuck>,
<https://tr.wikipedia.org/wiki/Brainfuck>)

Bu problemde basitleştirilmiş bir Brainfuck versiyonu için hafıza yönetimi yazmanız beklenmekte.

Sistemde 100 byte'lık başta 0 ile doldurulmuş bir hafızanız olduğunu varsayın. Bir pointer başlangıçta bu hafızanın ilk byte'ını gösteriyor olsun.

Basitleştirilmiş Brainfuck'ta 5 farklı komut bulunmakta:

> : Bu komut pointerınızın bir sonra hafıza lokasyonunu göstermesini sağlar.

< : Bu komut pointerınızın bir önceki hafıza lokasyonunu göstermesini sağlar.

+ : Bu komut pointerın gösterdiği byte'ı 1 artırmanızı sağlar.

- : Bu komut pointerın gösterdiği byte'ı 1 azaltmanızı sağlar.

. : Pointerın gösterdiği hafıza lokasyonundaki değeri ekrana bastırmanızı sağlar.

Bu problemde sizden beklenen, brainfuck komut dizisi verildiğinde, oluşacak hafızanın son durumunu ekrana basmanız. ('.' komutu geldiğinde ekrana birşey basmayın.)

Pointer ve hafızalardaki byte değerleri dairesel(circular)dir. Yani pointer 100. byte'ı gösteriyorsa, artırıldığında 1. byte'ı gösterir. 1. byte'ı gösteriyorsa, azaltıldığında 100. byte'ı gösterir.

Aynı şekilde hafıza değerleri de dairesel(circular)dir. 255(FF) byte'ını bir artırırsak 0(00) byte'ını, 0(00) byte'ını bir azaltırsak da 255(FF) byte'ını elde ederiz.

Input

Input birden fazla testten oluşacaktır.

Inputun ilk satırı test sayısını verir.(T)

Sonraki T satırın her birinde, bir adet brainfuck programı verilir.

1 <= T <= 100 1 <= PROGRAM_UZUNLUGU <= 100000

Output

Her test için hafızanın son durumunu 16'lık tabanda aralarında boşluk koyarak ekrana basınız. Her test outputunun başında "Case #: " formunda, test numarasını belirtiniz.(bkz. Örnek Output)

Örnek Input

5

```
<<<+++++><>-...-+
+>>>>+...+
+.-.+<><+
>>>+++>>>...>>>+-
+..<>..<>..+
```

Örnek Output

```
Case 1: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 04 FF 00
Case 2: 01 01 01 02 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00
Case 3: 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 01
Case 4: 00 00 00 03 00 00 00 00 00 00 01 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
Case 5: 02 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

Time Limit

C/C++/Java: 1 saniye, Python: 2 saniye