



Atingiu 1,00 de 1,00

Compressão de Strings

Johnny é um cara legal que gostava de carros e brincar com <u>algoritmos</u> de compressão. Ele está trabalhando em um projeto no qual tem que lidar com cadeias de caracteres extremamente grandes. O maior problema de Johnny nesse trabalho é que essas strings são grandes demais para manipular diretamente, então ele precisa de uma representação alternativa (menor) para a mesma <u>informação</u>. Johnny pensou em usar uma técnica bem conhecida para comprimir as strings: trocar ocorrências consecutivas de um mesmo caractere por uma única ocorrência deste mesmo caractere, seguida da quantidade de ocorrências. Neste formato, todo caractere é seguido por um inteiro positivo. Essa compressão permitiu que ele comunicasse suas strings, mas ele não consegue processá-las corretamente e agora precisa da sua ajuda para revertê-las às suas formas originais.

Entrada

A entrada é composta de uma linha contendo um inteiro N ($1 \le N \le 50$), seguida de N linhas distintas, cada uma com uma <u>string</u> codificada. É garantido que toda <u>string</u> tem pelo menos um caractere e está no formato comprimido, ou seja, é composto apenas por letras (maiúsculas) e dígitos.

Saída

Apresente as strings decodificadas, uma por linha.

Observações

• No caso de teste 1, devem ser decodificadas 4 strings. A primeira <u>string</u> tem 12 letras A, 2 letras B e 6 letras <u>C</u>, de forma que a <u>string</u> original é AAAAAAAAAAAAAAAAAAABBCCCCCC. A segunda <u>string</u> tem 3 letras A, 6 letras B e 8 letras F, de forma que a <u>string</u> original é AAABBBBBBFFFFFFF. A terceira <u>string</u> tem 16 letras A e 4 letras B, de forma que a <u>string</u> original é AAAAAAAAAAAAAAAAAABBBB. E a quarta <u>string</u> tem 5 letras G e 2 letras H, de forma que a <u>string</u> original é GGGGGHH.

For example:

Input	Result
4	AAAAAAAAAAABBCCCCCC
A12B2C6	AAABBBBBFFFFFFF
A3B6F8	AAAAAAAAAAAABBBB
A16B4	GGGGGHH
G5H2	
2	XAAAAAAAAADDDDDDDD
X1A10D10	VVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVV
V20	
5	AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAABBBB
A32B32C50A1V14G1B1	EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEGGGGGG
E30G90	DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD
D34	FFFFFFFFTTTTTTTTTTTTTTTTTTTTTTTTTTTTTTT
F9T45	KKKXXV
K3X2V1	

Answer: (penalty regime: 0, 0, 10, 20, ... %)

```
n = int(input())
 2 🔻
    for k in range(n):
 3
        numstr='0'
        carac = ''
 4
 5
        a = input()
        for letra in a:
 6
            if not(letra.isdigit()):
 7
 8
                 print(carac*int(numstr),end ="")
 9
                 carac = letra
                 numstr=''
10
11 1
12
                 numstr= numstr + letra
13
        print(carac*int(numstr))
14
15
```



PRECHECK VERIFICAR

	Input	Expected
~	4 A12B2C6 A3B6F8 A16B4 G5H2	AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
~	2 X1A10D10 V20	XAAAAAAAAAAADDDDDDDDD VVVVVVVVVVVVVVVVVV
~	5 A32B32C50A1V14G1B1 E30G90 D34 F9T45 K3X2V1	AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAABBBB
*	3 R2T2R2T2R2T2 F23G34 H1G1B1R2	RRTTRRTTRTT FFFFFFFFFFFFFFFFFFFGGGGGGGGGG
~	1 E1	Е
*	10 A1 B1 C1 D1 E1 F1 G1 H1 J1	A B C D E F G H I J
~	1 A1	A

Passou em todos os teste! ✔

Correto

Notas para este envio: 1,00/1,00.

