

Problem A: Sumando Numeros

Dados N números enteros, diseñar un algoritmo que retorne la suma de estos números.

Input

La entrada consta de varios casos de prueba. La primera línea contiene un entero T ($1 \leq T \leq 10$) denotando el número de casos de prueba.

Cada caso está compuesto por una línea que contiene el valor de N ($1 \leq N \leq 10$). La siguiente línea contendrá los N números enteros a_1, a_2, \dots, a_N que se deben de sumar ($0 \leq a_i \leq 10^3$).

Output

Para cada caso de prueba, el programa deberá imprimir "Caso # i : X " en una nueva línea donde i representa el número de caso de prueba y X la suma de los N números.

Example

Standard Input	Standard Output
2	Caso #1: 10
4	Caso #2: 17
1 2 3 4	
5	
3 5 0 5 4	

Solution

Solución en Python:

```
1 T = int(raw_input())
2 for iT in xrange(1, T + 1):
3     numbers = map(int, raw_input().split(' '))
4     print "Caso #%d: %d" % (iT, sum(numbers))
```

Judging Stage

El programa enviado al **juez** será evaluado con casos diversos casos de prueba y la salida será evaluada con el archivo de salida correcto que se encuentra cargado en el **juez**.

Por ejemplo, para el problema anterior se han cargado los archivos `Judge.in` y `Judge.out` indicando los datos de entrada y salida. El **juez** compilará el programa enviado y lo ejecutará con `Judge.in`, la salida `Answer.out` será comparada con `Judge.out` y si ambas salidas son iguales, el **juez** retornará un mensaje de **Accepted**.

Judge.in	Judge.out	Answer.out
10	Caso #1: 5490	Caso #1: 5490
10	Caso #2: 4361	Caso #2: 4361
896 545 866 525 713 764	Caso #3: 2338	Caso #3: 2338
265 159 473 284	Caso #4: 4858	Caso #4: 4858
9	Caso #5: 4060	Caso #5: 4060
528 489 971 691 244 222	Caso #6: 2358	Caso #6: 2358
526 81 609	Caso #7: 4776	Caso #7: 4776
5	Caso #8: 1511	Caso #8: 1511
839 480 486 26 507	Caso #9: 4608	Caso #9: 4608
9	Caso #10: 3280	Caso #10: 3280
941 97 399 555 330 212		
734 834 756		
9		
206 671 656 596 708 305		
282 0 636		
6		
547 91 120 942 48 610		
8		
756 584 554 663 474 309		
941 495		
3		
456 390 665		
9		
727 138 242 873 37 883		
984 63 661		
5		
900 972 283 488 637		