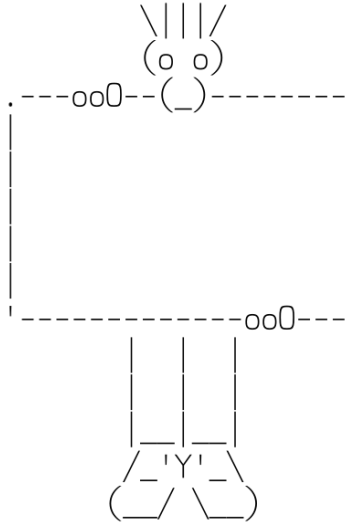


# Codificación binaria de Huffman

(tiempo límite: 1 segundo)

Todos conocemos el código ASCII, el cual define una correspondencia entre cierto conjunto de caracteres y un valor de 8 bits. Así por ejemplo, el carácter '?' corresponde al valor 00111111 (63 en decimal). Supongamos sin embargo que en un determinado aplicativo se utiliza un subconjunto de los 256 caracteres del ASCII y se conoce la frecuencia exacta de utilización de cada carácter.



Fuente: <https://freesvg.org/ascii-man-sign2-arvin61r58>

El problema consiste entonces en usar el código de Huffman ([http://es.wikipedia.org/wiki/Codificaci%C3%B3n\\_Huffman](http://es.wikipedia.org/wiki/Codificaci%C3%B3n_Huffman)) para determinar el valor en binario (no necesariamente de 8 bits) que debería tener cada carácter.

## Entrada

La entrada comienza con un valor entero positivo en una única línea, no mayor a 10 y que consiste en la cantidad de casos de prueba. Cada caso de prueba comienza con una línea que tiene la cantidad  $N$  de caracteres que tiene el alfabeto ( $2 \leq N \leq 256$ ). Luego siguen  $N$  líneas cada una dos valores separados entre sí por un espacio en blanco: el carácter  $C_i$  y su frecuencia  $F_i$  ( $0 < F_i < 100000$ ).

## Salida

Por cada caso de prueba se debe mostrar en una única línea el mensaje (sin las comillas): "caso i:". Posteriormente se deben mostrar, de a uno por línea, los caracteres correspondientes del caso, seguidos por un espacio en blanco, seguidos por el código binario resultante. Los caracteres deben mostrarse, no en el orden en que aparecen en la entrada, sino según el recorrido En-Orden del árbol binario resultante. Debe dejarse una línea vacía entre casos.

## Ejemplo de entrada

2  
6  
A 45  
B 13  
C 12  
D 16  
E 9  
F 5  
3  
X 9  
Y 8  
Z 7

### **Ejemplo de salida**

caso 1:

A 0  
C 100  
B 101  
F 1100  
E 1101  
D 111

caso 2:

X 0  
Z 10  
Y 11