

CAB²O

(tiempo límite: 1 segundo)

Seguro pasaste por alto el exponente y pensaste en el pingüino. No lo intentes negar. Bueno, en este problema en realidad CAB²O o CABBO se refiere a Costo del Árbol Binario de Búsqueda Óptimo, así que sácate de la cabeza “bonitos y gorditos”.

Se deben almacenar $\{e_1, e_2, \dots, e_N\}$ elementos numéricos diferentes en un árbol binario de búsqueda. Si cada elemento tiene una frecuencia entera no negativa $\{f_1, f_2, \dots, f_N\}$, y se cumple que $e_1 < e_2 < \dots < e_N$, cuál debe ser la estructura de dicho árbol de manera que se minimice el valor de:

$$T = \sum_{i=1}^N f_i * c(e_i)$$

Donde $c(e_i)$ es igual a uno más la cantidad de nodos en el camino simple que conecta la raíz con el elemento e_i (c de la raíz por tanto es uno).

Entrada

La entrada comienza con una línea que contiene un número entero C que corresponde a la cantidad de casos de prueba (no más de 20). Luego siguen C líneas, cada con N ($0 \leq N \leq 50$) valores correspondientes a las frecuencias de los elementos, no mayores a 10000 y separadas entre sí por un espacio en blanco.

Salida

La salida debe contener C líneas, cada una con el valor de T óptimo correspondiente.

Ejemplo de entrada

```
3
5
2 2 2
16 8 4 2 1
```

Ejemplo de salida

```
5
10
57
```