

# Soluciones entrenos OIE

## Haciendo Sushi

Takeshi Kitano está preparando sushi para sus  $n$  amigos. Sus amigos tienen gustos muy distintos y a cada uno le gusta un tipo de sushi diferente. Cada tipo de sushi usa  $g_i$  gramos de pescado. Para que haya más variedad Takeshi quiere comprar  $n$  tipos de pescado diferentes, uno para cada tipo de sushi. Cada pescado cuesta  $p_i$  yenes el gramo.

Dados los  $g_i$  y los  $p_i$ , cuál es el mínimo precio que tiene que pagar Takeshi Kitano para preparar sushi para sus amigos?

Cada tipo de sushi solo puede usar un tipo de pescado y se usan todos los tipos de pescado, uno para cada tipo de sushi.

### Input Format

La entrada empieza con un entero  $t$ , seguido de  $t$  casos.

Cada caso contiene una línea con un entero  $n$ . La segunda línea contiene los  $n$  enteros  $g_i$ . La tercera línea contiene los  $n$  enteros  $p_i$ .

### Constraints

$$1 \leq t \leq 100$$

$$1 \leq t \leq 10^5$$

$$n_{max} \cdot t \leq 2.5 \cdot 10^6$$

Donde  $n_{max}$  es el mayor valor de  $n$  en el fichero de entrada.

$$1 \leq g_i, p_i \leq 10^6$$

- 20 Puntos:  $1 \leq n \leq 10$
- 20 Puntos:  $1 \leq n \leq 10^2$
- 20 Puntos:  $1 \leq n \leq 10^3$
- 20 Puntos:  $1 \leq n \leq 10^4$
- 20 Puntos:  $1 \leq n \leq 10^5$

### Output format

Para cada caso escribid una línea con el precio mínimo que tiene que pagar Takeshi Kitano.

**Atención:** el resultado puede ser muy grande y no caber en un int, por eso hay que usar long long int en C++ o su equivalente en otros lenguajes.

## Solución

Para cada uno de los tipos de sushi, debemos escoger con que tipo de pescado lo hacemos, de modo que el precio total sea mínimo. La solución consiste en asignar al tipo de sushi que necesita más cantidad de pescado, aquel tipo de pescado que sea más barato, asignamos al segundo tipo de sushi que necesita más cantidad de pescado, el segundo tipo de pescado más barato, y vamos haciendo esto con cada tipo de sushi.

## Código

C++

```
1  #include <iostream>
2  #include <vector>    //para usar vectores
3  #include <algorithm> //para usar sort
4
5  using namespace std;
6  typedef long long int ll; //a partir de ahora, ll equivale a long long int
7
8  int main(){
9      int t;
10     cin >> t;
11
12     for (int caso = 0; caso < t; ++caso){
13         int n;
14         cin >> n;
15
16         vector <ll> G(n), P(n);
17
18         for (int i = 0; i < n; ++i) cin >> G[i]; //leemos las cantidades
19         for (int i = 0; i < n; ++i) cin >> P[i]; //leemos los precios
20
21         //ordenamos los dos vectores
22         sort(G.begin(), G.end());
23         sort(P.begin(), P.end());
24
25         ll solucion = 0;
26         for (int i = 0; i < n; ++i){
27             solucion += G[i] * P[n-i-1];
28         }
29         cout << solucion << endl;
30     }
31 }
```

## Python

```
1 t = int(input())
2
3 for caso in range(t):
4     n = int(input())
5     B = list(map(int, input().strip().split(' '))) #leemos las cantidades
6     P = list(map(int, input().strip().split(' '))) #leemos los precios
7
8     #ordenamos los dos vectores
9     B.sort()
10    P.sort()
11
12    solucion = 0
13    for i in range(n):
14        solucion += B[i] * P[n-i-1]
15
16    print(solucion)
```