
Estalviant electricitat**X86969_ca**

Teniu al vostre càrrec la gestió de la producció de farina en una indústria farinera. Cal produir d kilograms de farina, tasca per a la qual es disposa de n molins. En una hora, el molí i -èsim pot produir p_i kg de farina, i consumeix c_i kWh (kilowatts hora). A més, cada molí té una autonomia de u_i hores, i no pot estar en funcionament més temps per raons de manteniment. Amb els preus actuals de l'energia, és primordial minimitzar el consum elèctric. Podeu dissenyar un pla de producció òptim?

Per exemple, suposem que per a produir 3500 kg de farina tenim dos molins tals que:

- el primer en una hora produeix 50 kg de farina i consumeix 10 kWh, i pot funcionar fins a 20 hores.
- el segon en una hora produeix 75 kg de farina i consumeix 10 kWh, i pot funcionar fins a 40 hores.

Com que el segon molí consumeix el mateix que el primer però produeix més farina, aleshores un pla òptim consisteix en fer servir el segon molí durant 40 hores i el primer durant 10 hores, fabricant així $40 \cdot 75 + 10 \cdot 50 = 3500$ kg de farina amb un consum elèctric total de $40 \cdot 10 + 10 \cdot 10 = 500$ kWh.

Entrada

L'entrada consisteix en diversos casos. Cada cas comença amb n , el nombre de molins, seguit de n enters positius u_i , seguit de n enters positius p_i , seguit de n enters positius c_i , els quals representen l'autonomia, la productivitat i el consum de cadascun dels molins, respectivament. El cas acaba amb un enter d , que representa la demanda de farina. Podeu assumir que $1 \leq n \leq 10^4$, que $1 \leq u_i, p_i, c_i \leq 100$ per tot $0 \leq i \leq n - 1$, i que $0 \leq d \leq \sum_{i=0}^{n-1} u_i \cdot p_i$ (és a dir, segur que la demanda es pot satisfer).

Sortida

Per cada cas, escriviu amb quatre dígits decimals el consum mínim d'electricitat per tal de poder satisfer la demanda de farina.

Observació

Per escriure un `double` amb quatre dígits decimals podeu fer-ho així:

```
int main() {
    cout.setf(ios::fixed);
    cout.precision(4);
    double x;
    ...
    cout << x << endl;
```

Si us cal, podeu repassar com usar la funció `sort` per a ordenar un vector seguint un ordre ad-hoc aquí: <https://xuleta.jutge.org/stl/sort.html>

Exemple d'entrada 1

```
2
20 40
50 75
10 10
3500
```

```
3
1 3 2
5 2 3
7 3 4
10
```

```
4
1 2 1 2
1 2 3 4
2 3 4 5
14
```

Exemple de sortida 1

```
500.0000
13.6000
18.5000
```

Informació del problema

Autoria: Enric Rodríguez

Generació: 2026-01-25T16:57:16.079Z

© *Jutge.org*, 2006–2026.

<https://jutge.org>