

---

**Expressió aritmètica màxima****X62753\_ca**

---

Donada una seqüència de  $n$  nombres reals positius  $x_0, x_1, \dots, x_{n-1}$ , considerem les expressions aritmètiques que es poden formar amb sumes i productes usant *tots* aquests nombres (potser afegint parèntesis) *sense canviar-ne l'ordre relatiu*.

Per exemple, si  $n = 4$  i la seqüència és  $(0.1, 2.5, 3.0, 2.2)$ , aleshores algunes d'aquestes expressions són (ometem els parèntesis quan és possible per llegibilitat):

- $0.1 + 2.5 + 3.0 + 2.2$
- $0.1 \times 2.5 \times 3.0 \times 2.2$
- $0.1 \times 2.5 + 3.0 \times 2.2$
- $(0.1 + 2.5) \times 3.0 + 2.2$

El nostre objectiu és calcular el nombre més gran que es pot obtenir avaluant aquestes expressions. Per exemple, si la seqüència és  $(2, 3, 4)$ , aquest nombre és 24 i s'obté amb l'expressió  $2 \times 3 \times 4$ . Si en canvi la seqüència és  $(0.1, 2.5, 3, 2.2)$ , aleshores el màxim és 17.16 i s'obté amb  $(0.1 + 2.5) \times 3 \times 2.2$ .

**Entrada**

L'entrada consisteix en diversos casos. Cada cas comença amb  $n$ , seguit de  $x_0, x_1, \dots, x_{n-1}$ . Podeu assumir que  $n \geq 1$  i que  $x_i > 0$ .

**Sortida**

Per cada cas, escriviu amb quatre dígits decimals el nombre més gran que es pot obtenir avaluant les expressions que es poden formar amb sumes i productes i tots els nombres  $x_0, x_1, \dots, x_{n-1}$  sense canviar-ne l'ordre relatiu.

**Observació**

Resoleu aquest problema amb programació dinàmica.

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;
```

**Exemple d'entrada 1**

```
3 2.0 3.0 4.0
3 0.2 0.3 0.4
4 1.0 2.0 3.0 4.0
4 0.1 2.5 3.0 2.2
5 0.3 1.0 0.5 0.3 2.0
```

**Exemple de sortida 1**

```
24.0000
0.9000
36.0000
17.1600
4.2000
```

## **Informació del problema**

Autoria: Enric Rodríguez

Generació: 2026-01-25T17:21:27.702Z

© *Jutge.org*, 2006–2026.

<https://jutge.org>