

Squaremania 2 (squaremania2)

Marco si sta annoiando perché il gioco *Squaremania* gli risulta troppo semplice. Per fortuna viene a sapere che esiste una nuova versione del gioco: *Squaremania 2*.



Figura 1: L'icona di *Squaremania 2*.

Il gioco è simile: vengono forniti al giocatore N cubetti di legno 1×1 , ma a differenza della versione precedente non è obbligatorio formare un solo quadrato. L'obiettivo di *Squaremania 2* è quello di raggruppare i cubetti nel minor numero di quadrati possibile.

Ad esempio, se venissero forniti 13 cubetti, si potrebbe formare un quadrato 3×3 e uno 2×2 , e si può dimostrare che non esiste soluzione migliore.

Aiuta Marco trovando una soluzione ottimale al problema.

📖 Possono esistere più soluzioni ottimali. In tal caso, qualsiasi delle soluzioni ottimali verrà considerata corretta.

Implementazione

Dovrai sottoporre un unico file, con estensione `.cpp`.

📖 Tra gli allegati a questo task troverai un template `squaremania2.cpp` con un esempio di implementazione.

Il file di input è composto da 1 riga:

- Riga 1: l'intero N .

Il file di output è composto da 1 riga:

- Riga 1: Il numero di quadrati.




- Riga 2: Le lunghezze dei lati dei quadrati, separate da uno spazio.

Assunzioni

- $1 \leq N \leq 15\,000$.

Assegnazione del punteggio

Il tuo programma verrà testato su diversi test case raggruppati in subtask. Per ottenere il punteggio relativo ad un subtask, è necessario risolvere correttamente tutti i test che lo compongono.

- **Subtask 1** (0 punti) Casi d'esempio.

- **Subtask 2** (50 punti) $N \leq 11$

- **Subtask 3** (50 punti) Nessuna limitazione aggiuntiva.


Esempi di input/output

stdin	stdout
13	2 3 2
7	4 2 1 1 1
2789	2 50 17

Spiegazione

Il primo caso d'esempio è quello descritto nel testo del problema.

Nel secondo caso d'esempio, si possono formare 3 quadrati di dimensioni 1×1 e uno di dimensione 2×2 .

Nel terzo caso d'esempio conviene formare un quadrato 50×50 e uno 17×17 .