

Il libro del sus (bookpages)

Ultimamente Andrea si sta dedicando alla scrittura, e ha quasi finito di scrivere il suo capolavoro: *Sussone Fundamenta Prima*, un libro che raccoglie le tecniche base della programmazione che permettono al lettore di aprire la mente e di "Sussarsi" cit. *Sussone Fundamenta Prima*.

Per rendere il tomo più interessante, decide di numerare le pagine secondo un ordine particolare.



Figura 1: Il (presto) famoso libro di Andrea.

Il libro è composto da N pagine, numerate da 1 a N . Le pagine possono essere mostrate in ordine sparso (ossia è possibile ordinarle in un ordine diverso da $1, 2, \dots, N$). Andrea vuole sapere se è possibile creare un ordinamento tale che, prese 3 pagine consecutive del libro, i numeri delle pagine sommati siano divisibili per 3.

Ad esempio, se il libro avesse 6 pagine, potrebbero essere ordinate nel seguente modo:

$$[3, 2, 1, 6, 5, 4]$$

In modo tale che la proprietà valga per tutte le triplette consecutive:

- $3 + 2 + 1 = 3 \cdot 2$
- $2 + 1 + 6 = 3 \cdot 3$
- $1 + 6 + 5 = 3 \cdot 4$
- $6 + 5 + 4 = 3 \cdot 5$

Scrivi un ordinamento valido per le pagine del libro di Andrea!

👉 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 `bookpages.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: la risposta al problema.

Assunzioni

- $3 \leq N \leq 1\,000\,000$.
- Si può dimostrare che esiste sempre un ordinamento valido.

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** (30 punti) $N \leq 8$

- **Subtask 3** (50 punti) $N \leq 1\,000$

- **Subtask 4** (20 punti) Nessuna limitazione aggiuntiva.


Esempi di input/output

stdin	stdout
6	3 2 1 6 5 4
8	7 8 3 4 2 6 1 5
3	3 1 2

Spiegazione

Nei tre casi d'esempio, le permutazioni sono valide. Nota bene che non sono le uniche permutazioni valide, ad esempio nel primo caso anche la permutazione 2 1 6 5 4 3 è valida.