

algo2015

Piastrellature di un corridoio $2 \times n$ (piastrelle2)

Descrizione del problema

Abbiamo un corridoio di dimensione $2 \times N$ da piastellare. Sono a disposizione solamente piastrelle di dimensioni 1×1 e 1×2 ; queste ultime possono essere ruotate di 90 gradi. Vorremmo conoscere tutte le possibili disposizioni delle piastrelle sul pavimento del corridoio $2 \times N$. Lista le possibili piastrellature una alla volta. Dato che ogni piastrella sarà adiacente ad al più 4 piastrelle, risulta possibile labellare le piastrelle con i colori $\{1, 2, 3, 4, 5\}$ in modo che piastrelle adiacenti abbiano colore diverso. Per specificare una singola piastrellatura del corridoio (tiling della griglia $2 \times N$ con pezzi di domino, ossia con sottogriglie 1×2 oppure 2×1), specifica solo il colore per ciascuna delle $2N$ celle del corridoio.

File di input

Il programma deve leggere da un file di nome `input.txt` un unico intero N , che determina la dimensione del corridoio.

File di output

Il programma deve scrivere in un file di nome `output.txt`. La prima riga del file contiene un intero K , il numero di possibili piastrellature. Il file va organizzato in $3K$ righe. Per ogni $i = 1, \dots, K$, le righe $3i - 1$ e $3i$ rappresentano l' i -esima piastrellatura, e la riga successiva a queste serve come spaziatura. Per specificare l' i -esima piastrellatura, si prenda un qualsiasi 5-coloring della stessa e lo si riporti sulle due righe nel modo ovvio (rispettando cioè la topologia sottostante).

Assunzioni

- tempo massimo: 2 secondi.

Subtask

- **Subtask 1 [5 punti]:** caso di esempio.
- **Subtask 2 [15 punti]:** $N \leq 5$.
- **Subtask 3 [30 punti]:** $N \leq 8$.
- **Subtask 4 [40 punti]:** $N \leq 9$.
- **Subtask 5 [10 punti]:** $N \leq 10$.

Esempio di input/output

File input.txt	File output.txt
3	22 1 2 1 2 1 2 1 2 3 2 1 3 1 2 3 2 1 1 1 2 2 2 1 3 1 2 2 2 1 1 1 3 1 2 3 2 1 3 1 2 3 1 1 3 1 2 2 3 1 3 1 2 2 1 1 3 3 2 2 1 1 1 2 2 3 1 1 1 2 2 3 2 1 1 2 2 3 3 1 1 2 2 2 1 1 1 3 2 2 3 1 2 1 1 3 2 1 2 1 1 3 1 1 2 1 1 3 3 1 2 2 1 3 1 1 2 2 1 3 3 1 2 1 1 2 3 1 2 1 1 2 1