

C. Monster-Go

Nome del problema	Monster-Go
Limite di tempo	1 secondo
Limite di memoria	1 gigaottetto

Helen e i suoi amici hanno scoperto un nuovo fantastico gioco per i loro cellulari. Il gioco, chiamato *Monster-Go*, consiste nel catturare mostri camminando verso diversi nidi di mostri all'aperto. In ogni nido è disponibile un numero infinito di mostri dello stesso tipo. Quando gli amici arrivano al nido di un mostro, ognuno di loro lo cattura e lo aggiunge alla propria collezione. Ci sono in totale 50 mostri diversi che gli amici possono catturare, numerati $0, 1, \dots, 49$.

Per rendere il gioco più emozionante, gli N amici hanno deciso che ogni giocatore avrà una lista personalizzata di esattamente 12 tipi di mostri da collezionare. Il primo che cattura tutti i mostri presenti nella sua lista vince la partita. Vogliono strutturare le liste in modo tale che, indipendentemente dall'ordine in cui visitano i nidi dei mostri, ci sia sempre un singolo e univoco vincitore, mai un pareggio. Gli amici vanno sempre in giro insieme in gruppo e arrivano insieme al nido del mostro.

Puoi aiutarli a progettare le liste? Il tuo punteggio dipenderà dal numero di valori di N , ovvero dal numero di persone che giocano, per i quali sarai in grado di risolvere il problema.

Input

La prima e unica riga di input contiene l'intero N , il numero di giocatori.

Output

Devi stampare N righe, dove la i -esima riga contiene i 12 interi distinti $c_{i,1}, c_{i,2}, \dots, c_{i,12}$ (dove $0 \leq c_{i,j} \leq 49$) rappresenta i mostri nell'elenco della persona i . Se esistono più soluzioni, puoi stamparne una qualsiasi.

Assunzioni e punteggio

- $1 \leq N \leq 50$.

La tua soluzione verrà testata su una serie di subtask, ognuno dei quali vale un certo numero di punti. **L' i esimo subtask contiene un singolo caso di test con $N = i$ e vale 2 punti.** Vale a dire, ci sono un totale di 50 subtask (uno per ogni $N = 1, 2, \dots, 50$) e il tuo punteggio in questo problema è il doppio del numero di subtask risolti dal tuo programma.

Subtask	Punteggio	Assunzioni
1	2	$N = 1$
2	2	$N = 2$
3	2	$N = 3$
\vdots	\vdots	\vdots
49	2	$N = 49$
50	2	$N = 50$

Esempi

Nell'esempio, dove ci sono $N = 2$ amici, il programma dovrebbe produrre due liste. Infatti, per le due liste nell'output dell'esempio, gli amici non possono vincere entrambi contemporaneamente, indipendentemente dall'ordine in cui visitano i nidi dei mostri. Tieni presente che ci sono molte altre risposte valide.

Input	Output
2	<pre>0 1 2 3 4 5 6 7 8 9 10 11 38 39 40 41 42 43 44 45 46 47 48 49</pre>