Tiling Madness (`tiling`)

Après une longue nuit au « camp SOI », les participants sont devenus inexplicablement neuneus. Ils ont commencé à s'appeler « brudis » les uns les autres et ont découvert ces trucs bizarres appelés « Maos ». Ce sont des pièces formées de carrés, et tu vas devoir jouer avec.

Ton but est de paver une grille de $N \times N$ avec N copies identiques d'un « 2N-Mao », sans qu'elles se chevauchent.

Les 2N-Maos n'ont pas besoin d'être entièrement à l'intérieur de la grille $N \times N$.

Plus formellement, chaque solution au problème doit fixer un 2N-Mao, puis en placer N copies sur une grille (sans les tourner ni faire de symétrie) de sorte que :

- chaque case de la grille fait partie d'au plus un des 2N-Maos.
- il existe une sous-grille $N \times N$ entièrement recouverte par les 2N-Maos.

Un 2N-Mao est un ensemble connexe de $2N$ carrés ; tu peux trouver un exemple de 2N-Mao valide et d'un invalide dans la Fig. 1.

A	A	A	.
.	.	A	A
A	A	A	A
A	A	A	.
A	.	A	.

.	A	A	A	.
A	.	.	.	.
.	A	A	A	A
.	A	A	A	A
.	A	.	.	A

Fig. 1. – La figure de gauche est un 14-Mao valide. Celle de droite ne l'est pas, car elle n'est pas connexe.

On veut connaître plein de manières de paver la grille, chacune utilisant un 2N-Mao **unique**. Ton score dépendra du nombre de 2N-Maos valides que tu fournis et qui permettent de paver le carré $N \times N$. Plus tu trouves de solutions, plus les brudis deviendront des « lutins joyeux » (« happy imps ») !

Note que les 2N-Maos qui peuvent être obtenus l'un de l'autre par rotation ou symétrie sont considérés comme **distincts**.

Implémentation

C'est un problème de type « output-only ». Tu devras soumettre exactement un fichier de sortie.

Format de l'entrée

Le seul fichier d'entrée est composé d'une seule ligne, contenant l'entier N.

Format de la sortie

Le seul fichier de sortie doit être dans le format suivant :

- La première ligne doit contenir un unique entier C ($0 \leq C \leq 16000$) : le nombre de solutions différentes contenues dans ta sortie.
- Ensuite, C blocs de solution doivent suivre. Chaque bloc doit être dans le format suivant :

- La première ligne doit contenir deux entiers h et w ($0 \le h, w \le 5N$) : la hauteur et la largeur de la grille où tu vas placer les 2N-Maos.
- Les h lignes suivantes doivent chacune contenir une chaîne de caractères de longueur w, composée des N premières lettres majuscules de l'alphabet latin et du caractère point (`.`). La i-ème lettre de l'alphabet indique que la case est occupée par la i-ème copie du 2N-Mao, tandis que le point indique que la case est laissée vide.

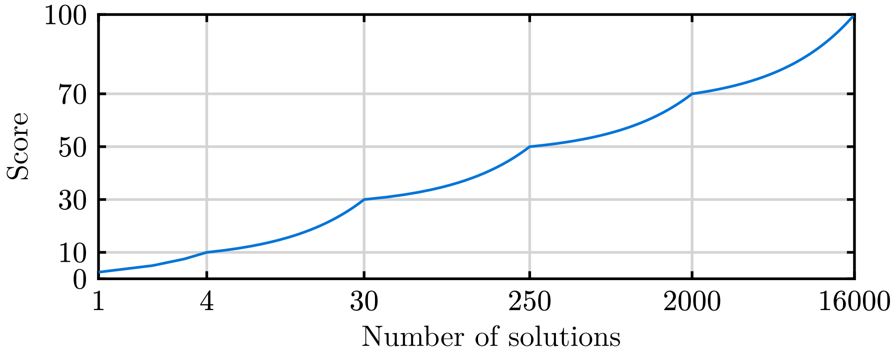
Pour chaque bloc de solution, la grille doit contenir une sous-grille $N \times N$ qui ne contient aucun caractère `.`. Toutes les N copies du 2N-Mao doivent être identiques.

Score

Ce problème n'a qu'un seul cas de test, où $N = 7$.

Le score S pour ta solution est déterminé selon le tableau suivant. Entre les valeurs spécifiées dans le tableau, le score sera attribué par **interpolation linéaire**. Une sortie mal formatée te donnera toujours zéro point.

Solutions	Score
0	0
4	10
30	30
250	50
2000	70
16000	100



Exemples

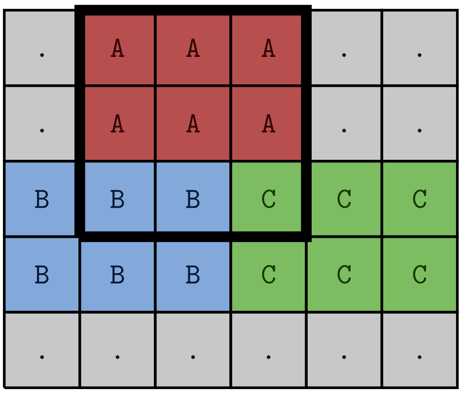
input	output

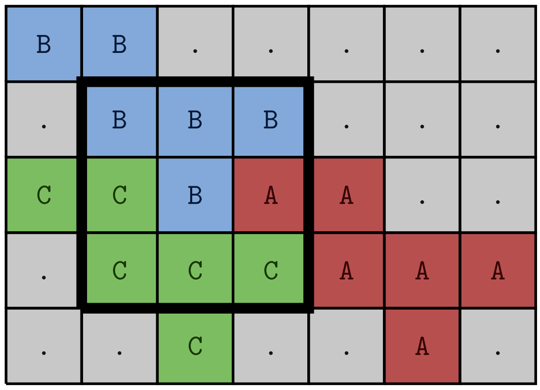
`3`	`2`
	`5 6`
	`.AAA..`
	`.AAA..`
	`BBBCCC`
	`BBBCCC`
	`......`
	`5 7`
	`BB.....`
	`.BBB...`
	`CCBAA..`
	`.CCCAAA`
	`..C..A.`

Explication

Dans **l'exemple** on te demande d'utiliser des 6-Maos pour couvrir un carré de 3×3 : note que ce n'est pas une entrée valide, car dans la seule entrée du problème, $N = 7$.

La sortie montre deux des nombreuses solutions possibles, illustrées dans l'image ci-dessous.





Dans les deux cas, on peut voir qu'il y a 3 6-Maos identiques qui ne se chevauchent pas et qu'un carré 3×3 est couvert.