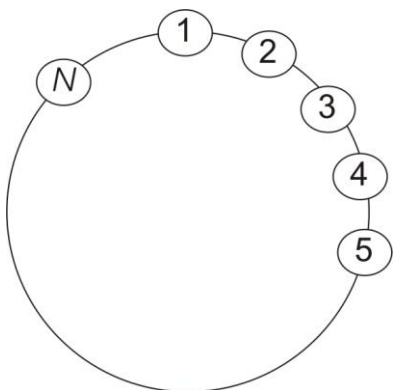


Завдання III етапу Всеукраїнської учнівської олімпіади з інформатики 19 лютого 2005 року

Завдання 1. *Наметове містечко.*



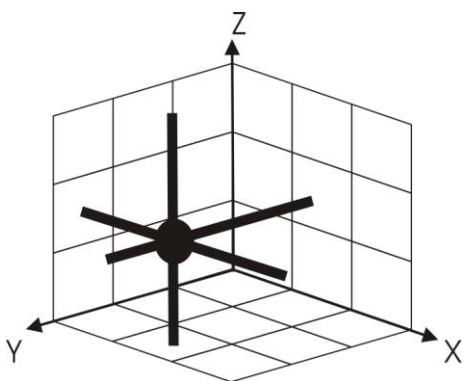
Після завершення Помаранчевої революції, комендант наметового містечка вирішив видати особливу грамоту «Стояв до останнього» тому, хто останнім зніме свій намет з Хрещатика.

Оскільки, бажаючих отримати єдину грамоту було багато, комендант вирішив діяти чесно. Він по колу розмістив кружечки з номерами усіх N наметів і попросив кожного члена наметового містечка задумати якесь число. Коли всі задумані числа додали, комендант отримав число M .

Далі комендант почав відраховувати від першого кружечка до кружечка з номером M . Номер намету, на який випадало число M знімався з колу, знімався сам намет, а комендант продовжував знову рахувати до M починаючи з наступного кружечка.

Допоможіть коменданту за заданими з клавіатури N і M встановити кому видавати грамоту.

Завдання 2. *Супер-шахи.*



Нами сконструйована нова гра – об'ємні шахи (називатимемо їх супер-шахи, і відповідно в них користуватимемося супер-фігурами і супер-дошкою).

Супер-дошка – куб розміром $N*N*N$ одиничних кубиків. Супер-фігура розміщується всередині будь-якого одиничного кубика.

Положення фігури описується на супер-дошці трьома координатами: (x,y,z) ; $1 \leq x \leq N$; $1 \leq y \leq N$; $1 \leq z \leq N$.

Правила пересування і «правила бою» супер-фігур схожі на звичайні правила.

Наприклад, супер-тура може бити будь-яку супер-фігуру, яка розміщена на одній прямій з нею, якщо ця пряма паралельна до одного з ребер одиничного кубика, в якому тура розміщена (так як це зображено на малюнку).

Скласти програму, яка б за заданим N виводила мінімальну кількість супер-тур, які можна розмістити на супер-дошці, щоб всі поля знаходилися під боєм і вивести хоча б одне з таких розміщень.

Формат вхідних даних

N задається з клавіатури ($N < 100$).

Формат вихідних даних

Результат виводиться у файл *chess.txt*, у якому в першому рядку число k – мінімальна кількість супер-тур, а в наступних k рядках по три числа розділених комами – координати кожної з супер-тури.

Завдання 3. Анаграматор.

Анаграматор — спеціальний пристрій для отримання зі слова його анаграм (тобто слів, які записані тими ж буквами, але в іншому порядку). Цей пристрій уміє виконувати 2 операції:

- 1) Взяти чергову букву вихідного слова і помістити її у стек;
- 2) Взяти букву зі стеку і додати її у кінець вихідного слова;

Стек — це вмістилище даних, яке працює за принципом «перший прийшов – останній пішов». Стек можна уявити собі як дитячу пірамідку. Коли ми додаємо букву у стек, це відповідає тому, що на стержень пірамідки ми надіваємо зверху кільце, на якому написано відповідну букву. Коли беремо букву зі стеку, то це відповідає тому, що ми знімаємо зі стержня верхнє кільце і дивимося яка буква на ньому написана.

Наприклад, слово TROT у слово TORT може бути перетворене Анаграматором двома різними послідовностями операцій: 11112222 або 12112212.

Напишіть програму, яка за двома заданими словами обчислює кількість різних послідовностей операцій Анаграматора, які перетворюють перше з цих слів у друге, а також знаходить самі ці послідовності.

Формат вхідних даних

Перша стрічка вхідного файлу *an_in.txt* містить вхідне слово, а друга — слово, яке слід отримати. Слова містять тільки великі латинські букви і мають довжину не більше 50 символів. Обидва слова мають однакову довжину. В цих строках не міститься пробілів.

Формат вихідних даних

У першій стрічці вихідного файлу *an_out.txt* виводиться кількість послідовностей операцій Анаграматора з допомогою яких можна перетворити перше слово у друге, а також самі послідовності операцій. Кожна послідовність повинна бути виведена в окремій стрічці і містити лише цифри 1 і 2 (які вказують порядок виконання операцій), виведених без пробілів.

Приклади

<i>an_in.txt</i>	<i>an_out.txt</i>
MADAM ADAMM	1111222122 1111222212 1121212122 1121212212 4
ADAMM MADAM	1111222212 1
LONG GONG	0
AAAAA AAAAA	11111122222 42