Преферанс

Имя входного файла:	`pref.in`
Имя выходного файла:	`pref.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайта

В новой колоде 32 карты для преферанса расположены в следующем порядке (сверху вниз): червы, бубны, трефы, пики. В каждой масти сначала лежит семерка, под ней — восьмерка, затем девятка, десятка, валет, дама, король, туз. Тасовка карт осуществляется так: 16 карт, составляющих верхнюю половину колоды, распределяются между картами нижней половины колоды. Каждая карта верхней половины вставляется в нижнюю колоду так, что в получившейся колоде карты верхней половины идут в том же порядке, в котором они были изначально. Любое число карт верхней половины можно располагать как над верхней, так и под нижней картой второй половины колоды, а также между любыми двумя соседними картами нижней половины колоды. Такие действия повторяются не более пяти раз.

Требуется написать программу, которая указывает, как надо осуществить тасовку, чтобы в итоге получить заранее заданное расположение карт.

Формат входного файла

Единственная строка входного файла содержит информацию о порядке карт, в котором они должны оказаться после тасовки. Карты перечислены сверху вниз.

Каждая карта обозначается латинской буквой, указывающей масть (пики — $\widehat{\mathrm{S}}$, трефы — $\widehat{\mathrm{C}}$, бубны — $\widehat{\mathrm{D}}$, червы — $\widehat{\mathrm{H}}$), и номиналом (туз — $\widehat{\mathrm{A}}$, король — $\widehat{\mathrm{K}}$, дама — $\widehat{\mathrm{Q}}$, валет — $\widehat{\mathrm{J}}$, десятка — $\widehat{0}$, остальные — в соответствии со своим значением: $\widehat{9}$, $\widehat{8}$, $\widehat{7}$).

Формат выходного файла

В первую строку выходного файла необходимо вывести целое число N ($0 \le N \le 5$) — количество шагов тасовки. Следующие N строк должны содержать информацию о каждом шаге тасовки. Каждая строка при этом должна содержать 16 чисел, указывающих номера позиций, на которых оказываются снятые карты. Номера позиций выводятся в порядке возрастания и разделены пробелами. Нумерация позиций производится сверху вниз от 1 до 32.

Пример

В примере входная строка для удобства разбита на две. Во входных файлах при тестировании задачи будет ровно одна строка, завершенная переводом строки.

pref.in
C7 H7 C8 H8 C9 H9 C0 H0 S7 D7 S8 D8 S9 D9 S0 D0 SJ DJ SQ DQ SK DK SA DA CJ HJ CQ HQ CK HK CA HA

pref.out
2 2 4 6 8 10 12 14 16 18 20 22 24 26 28 30 32 1 2 3 4 5 6 7 8 25 26 27 28 29 30 31 32

Авторегистратор

Имя входного файла:	`autoreg.in`
Имя выходного файла:	`autoreg.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайт

Повторение, мать...

Любитель длинных паролей

Для регистрации участников соревнований использовалась передовая система RHR («Raven-Hedgehog Registrator»). После 239-й попытки повторного ввода пароля команды Гвидо, которому Скив поручил зарегистрировать их команду, сдался.

— Не знаю, в чём тут дело. Я ввожу подтверждение пароля, после чего система просто зависает.

— Сейчас, выясним... Ааз через какое-то хитрое устройство связался с оргкомитетом.

Оказалось, что в системе RHR пароли шифруются добавлением к строке пароля спереди и сзади произвольных символов. При этом ответственный за регистрацию (в RHR традиционно используется ручная обработка информации) почему-то получает на экран строку с зашифрованными паролем и зашифрованным подтверждением, да еще и соптимизированную нетрадиционным методом для экономии сетевого трафика. Так что разобраться, был ли пароль повторён правильно, ответственный был не в состоянии.

Ааз предложил следующий способ: если во входной строке содержатся две одинаковые (возможно, частично пересекающиеся) подстроки заданной длины k (рекомендуемая длина пароля), то считать, что процесс ввода пароля завершён успешно.

Для контроля ответственного за регистрацию напишите программу, которая автоматически определяет успешность завершения процесса ввода пароля в системе RHR.

Формат входного файла

В первой строке входного файла находится непустая строка S длиной не более 1 000 000 символов, состоящая из маленьких и больших букв латинского алфавита, при этом большие и маленькие буквы различаются. Во второй строке дано число k ($1 \le k \le \mathrm{length}(S)$).

Формат выходного файла

В единственную строку выходного файла выведите «$\widehat{\mathrm{Y}}$ES», если процесс ввода пароля завершён успешно, и «$\widehat{\mathrm{N}}$O», если нет.

Пример

autoreg.in	autoreg.out
sdkfgIUDHFSdifuhdsPIFSUDiu 1	YES

Мега-инверсии

Имя входного файла:	`mega.in`
Имя выходного файла:	`mega.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайта

Инверсией в перестановке $p_1, p_2, ..., p_N$ называется пара (i, j) такая, что $i < j$ и $p_i > p_j$. Назовем мега-инверсией в перестановке $p_1, p_2, ..., p_N$ тройку (i, j, k) такую, что $i < j < k$ и $p_i > p_j > p_k$. Придумайте алгоритм для быстрого подсчета количества мега-инверсий в перестановке.

Формат входного файла

Первая строка входного файла содержит целое число N ($1 \le N \le 100\,000$). Следующие N чисел описывают перестановку: $p_1, p_2, ..., p_N$ ($1 \le p_i \le N$), все p_i попарно различны. Числа разделяются пробелами и/или переводами строк.

Формат выходного файла

Единственная строка выходного файла должна содержать одно число, равное количеству мега-инверсий в перестановке $p_1, p_2, ..., p_N$.

Примеры

mega.in	mega.out
4 4 3 2 1	4