Problem A. Baggage

Input:	`Standard`
Output:	`Standard`
Judge:	`Kattis`

An airline has two flights leaving at about the same time from ICPCity, one to city B and one to city A. The airline also has n counters where passengers check their baggage. At each counter there is a pair of identical baggage bins, one for city B and one for city A.

Just before the flights depart, each pair of baggage bins is moved by a motorized cart to a sorting area. The cart always moves two bins at a time, one for city B and one for city A. After all the bins have been moved, they line up in the sorting area like this:

B A B A B A ... B A

That is, there are $2n$ baggage bins in a row, starting with a bin for city B, then one for city A, and so forth. The task now is to reorder them so all the baggage bins for city A precede the baggage bins for city B. Then the bins can be loaded on the appropriate aircraft.

The reordering is done by moving pairs of adjacent baggage bins (not necessarily B then A), again via the motorized cart. For proper balance, the cart must always carry two bins, never just one. A pair of bins must always be moved to an empty space that is at least two bins wide. On the left of the first bin are some empty spaces that can be used as needed during the reordering.

When the reordering process begins, the bin locations are numbered from 1 (initially containing the leftmost B baggage bin) to $2n$ (initially containing the rightmost A baggage bin). There are $2n$ initially empty spaces to the left of the bins, numbered from 0 to $-2n+1$, as shown in the figure for the case $n = 4$.

								B	A	B	A	B	A	B	A
-7	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6	7	8

Given n, find a shortest sequence of moves that will reorder the bins so that all the A bins are to the left of all the B bins. At the end of the process, it is possible that the leftmost A bin is at some location other than 1, but the bins must be adjacent in a sequence of $2n$ locations.

Input

The input consists of a single test case, which consists of the integer n $(3 \leq n \leq 100)$.

Output

Display a shortest sequence of moves that will correctly reorder the bins. Each move is of the form "f to t", where f and t are integers representing the movement of the bins in locations f and $f+1$ to locations t and $t+1$. If multiple solutions are possible, display any one of them.

Samples

Input	Output
5	8 to -1 3 to 8 6 to 3 0 to 6 9 to 0

Input	Output
8	10 to -1 3 to 10 14 to 3 7 to 14 0 to 7 11 to 0 4 to 11 15 to 4

URL:
https://icpc.kattis.com/problems/baggage

Problem B. Caesar's Legions

Input:	`Standard`
Output:	`Standard`
Judge:	`CodeForces`

Gaius Julius Caesar, a famous general, loved to line up his soldiers. Overall the army had n_1 footmen and n_2 horsemen. Caesar thought that an arrangement is not beautiful if somewhere in the line there are strictly more that k_1 footmen standing successively one after another, or there are strictly more than k_2 horsemen standing successively one after another. Find the number of beautiful arrangements of the soldiers.

Note that all $n_1 + n_2$ warriors should be present at each arrangement. All footmen are considered indistinguishable among themselves. Similarly, all horsemen are considered indistinguishable among themselves.

Input

The only line contains four space-separated integers n_1, n_2, k_1, k_2 $(1 \le n_1, n_2 \le 100, 1 \le k_1, k_2 \le 10)$ which represent how many footmen and horsemen there are and the largest acceptable number of footmen and horsemen standing in succession, correspondingly.

Output

Print the number of beautiful arrangements of the army modulo 100000000 (10^8). That is, print the number of such ways to line up the soldiers, that no more than k_1 footmen stand successively, and no more than k_2 horsemen stand successively.

Samples

Input	Output
`2 1 1 10`	`1`
`2 3 1 2`	`5`
`2 4 1 1`	`0`

Note: Let's mark a footman as 1, and a horseman as 2.

In the first sample the only beautiful line-up is: 121.

In the second sample 5 beautiful line-ups exist: 12122, 12212, 21212, 21221, 22121.

URL:
http://codeforces.com/problemset/problem/118/D

Problem C. Edge case

Input:	`Standard`
Output:	`Standard`
Judge:	`UVa`

In graph theory, a *matching* or *independent edge set* in a graph $G = (V, E)$ is a set of edges $M \subseteq E$ such that no two edges in the matching M share a common vertex.

Recently you saw in the news that "The Sveriges Riksbank Prize in Economic Sciences in Memory of Alfred Nobel" (informally, the Nobel Prize in Economics) for 2012 was awarded to Alvin E. Roth and Lloyd S. Shapley for, amongst other things, their algorithm for finding a matching satisfying certain criteria in a bipartite graph. Since you have also heard that matchings in *cycle graphs* have applications in chemistry your thoughts centre around a plan for a beautiful future where your Christmas shopping is more luxurious than ever!

The cycle graph, C_n, $n \geq 3$, is a simple undirected graph, on vertex set $\{1, \ldots, n\}$, with edge set $E(C_n) = \{\{a, b\} \mid |a - b| \equiv 1 \ mod \ n\}$. It is 2-regular, and contains n edges. The graphs C_3, C_4, C_5, and C_6 are depicted in Figure 1.

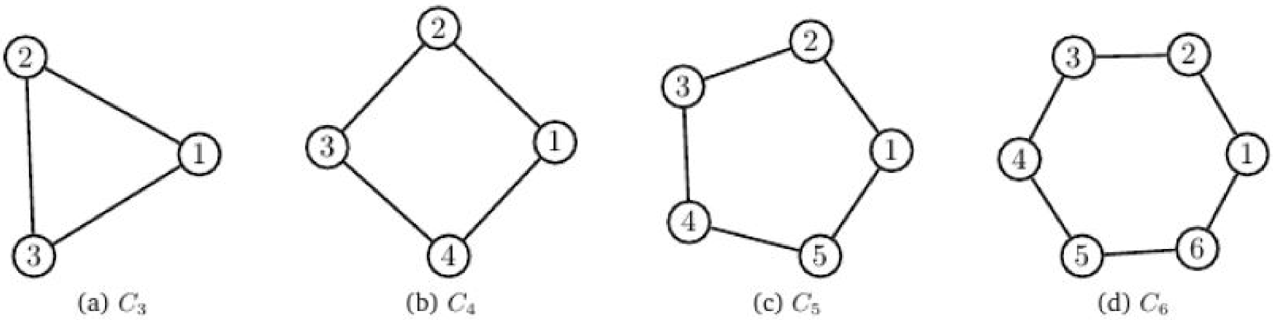


Figure 1: The graphs C_3, C_4, C_5, and C_6.

Your first step towards Nobel Prize fame is to be able to compute the number of matchings in the cycle graph C_n. In Figure 2 the seven matchings of the graph C_4 are depicted.

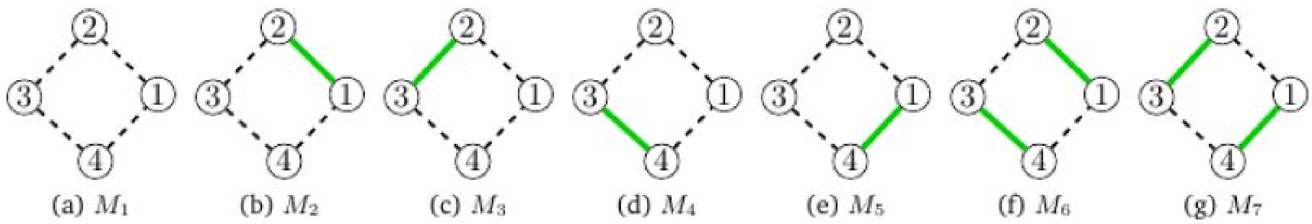


Figure 2: The matchings of C_4 . The edges that are part of the respective matching are coloured green, while the edges left out of the matching are dashed. $M_1 = 0$, $M_2 = \{\{2, 1\}\}$, $M_3 = \{\{3, 2\}\}$, $M_4 = \{\{4, 3\}\}$, $M_5 = \{\{1, 4\}\}$, $M_6 = \{\{2, 1\}, \{4, 3\}\}$, and $M_7 = \{\{3, 2\}, \{1, 4\}\}$.

Input

For each test case, you get a single line containing one positive integer: n, with $3 \leq n \leq 10000$.

Output

For each test case, a row containing the number of matchings in C_n.

Samples

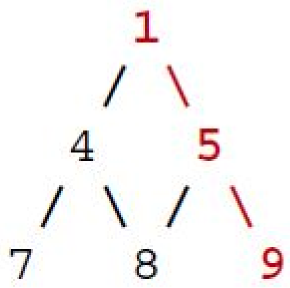
Input	Output
3 4 100	4 7 792070839848372253127

URL:
http://uva.onlinejudge.org/index.php?
option=onlinejudge&page=show_problem&problem=4521

Problem D. Two Mysterious Alphabets from a Tree

Input:	`Standard`
Output:	`Standard`
Judge:	`COJ`

Your task is to extract 2 alphabets from a binary tree which is composed of unsigned integers respecting the following rules. Let n be the height of a tree. At the level k ($1 \leq k \leq n$), the tree contains k of nodes and each node has 2 children nodes (except the leaf nodes at the level n which have no children). See the example below to understand the tree formation. Some nodes may have 2 parent nodes.



You need to walk in a tree on the path that has a maximum summation (e.g., $1 + 5 + 9 = 15$). Numbers in each summation cannot cross into different links (e.g., 5+7 is illegal). Then, your intermediate task is to calculate 2 numbers for alphabet extraction. The first number is calculated from $\sum_{i=1}^{n} i^2$ where i is a number along the maximum summation path and n is the height of a tree. The second number is a summation of the maximum path $\sum_{i=1}^{n} i$. Regarding to the example above, the first number $= 1 + 25 + 81 = 107$ and the second number $= 1 + 5 + 9 = 15$.

Finally, these two numbers are transformed into two lower case alphabets from "a" to "z" respectively, where "a" is used for 0 and "z" is used for 25. Since there are only 26 alphabets, a number greater than 25 will reuse the same set of alphabets. For example, 107 = "d" and 15 = "p" (that is, the first alphabet "a" = 0, or 26, or 52 etc).

Write a program to find the 2 mysterious alphabets from a given tree.

Input

The first line of input contains the height (n) of a tree ($0 \leq n \leq 100$). The second line contains unsigned integer numbers (i) in each level of a tree ($0 \leq i \leq 100$), consecutively. Assume that there is only one maximum path in a tree.

Output

The first line contains two integer calculated from the rules above, and the second line contains 2 decoded alphabets.

Samples

Input	Output
3 1 4 5 7 8 9 4 1 5 2 5 1 9 3 4 20 1 5 2 4 9 1 3 1 1 1 1 2 12 5 4 3 2 6 9 8 8 7 7 7 9 1 1 3 8 2 10 5 1 2 3 2 1 9 81	107 15 dp 486 32 sg 166 20 ku 6765 109 ff

URL:
http://coj.uci.cu/24h/problem.xhtml?pid=2632

Credits

LaTeX edition by:

Daniel Cañizares Corrales.
Professor, Universidad Católica de Oriente.

All problems can be found on the provided URLs.
This compilation was made for educational purposes.