Greetings From Globussoft

- Given below are 5 Programming questions, you have to solve any 3 out of 5 questions.

- These 5 questions you can attempt in any technology like C/C++, java, .Net, PHP

- To solve these 3 questions you’ve max. 3 hours.

- While Solving these questions you are not allowed to use any **Search Engine** like Google, Yahoo, Bing …

All the best for your test

Globussoft

QUESTION - 1

"Hike on a Graph" is a game that is played on a board on which an undirected graph is drawn. The graph is complete and has all loops, i.e. for any two locations there is exactly one arrow between them. The arrows are coloured. There are three players, and each of them has a piece. At the beginning of the game, the three pieces are in fixed locations on the graph. In turn, the players may do a move. A move consists of moving one's own piece along an arrow to a new location on the board. The following constraint is imposed on this: the piece may only be moved along arrows of the same colour as the arrow between the two opponents' pieces.

In the sixties ("make love not war") a one-person variant of the game emerged. In this variant one person moves all the three pieces, not necessarily one after the other, but of course only one at a time. Goal of this game is to get all pieces onto the same location, using as few moves as possible. Find out the smallest number of moves that is necessary to get all three pieces onto the same location, for a given board layout and starting positions.

Input

The input file contains several test cases. Each test case starts with the number n. Input is terminated by $n=0$. Otherwise, $1<=n<=50$. Then follow three integers p_1, p_2, p_3 with $1<=p_i<=n$ denoting the starting locations of the game pieces. The colours of the arrows are given next as a $n \times n$ matrix m of whitespace-separated lower-case letters. The element m_{ij} denotes the colour of the arrow between the locations i and j. Since the graph is undirected, you can assume the matrix to be symmetrical.

Output

For each test case output on a single line the minimum number of moves required to get all three pieces onto the same location, or the word "impossible" if that is not possible for the given board and starting locations.

Example

Input:

```
3 1 2 3
r b r
b b b
r b r
2 1 2 2
y g
g y
0
```

Output:

```
2
impossible
```

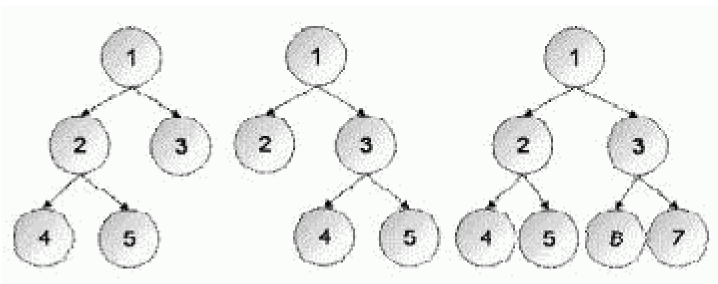
QUESTION – 2

Tree is an important data structure in Computer Science. Of all trees we work with, Binary Tree is probably the most popular one. A Binary Tree is called a **Strictly Binary Tree** if every nonleaf node in a binary tree has nonempty left and right subtrees. Let us define **a Strictly Binary Tree of depth d**, as a Strictly Binary Tree that has at least one root to leaf path of length **d**, and no root to leaf path in that tree is longer than **d**. So let us use a similar reasoning to define a generalized structure.

An n-ary Tree is called a **Strictly n-ary Tree** if every nonleaf node in an n-ary tree has n children each. A **Strictly n-ary Tree of depth d** can now be defined as a Strictly n-ary Tree that has at least one root to leaf path of length d, and no root to leaf path in that tree is longer than d.

Given the value of n and depth d, your task is to find the number of different strictly n-ary trees of depth d.

The figure below shows the 3 different strictly binary trees of depth 2.



Input

Input consists of several test cases. Each test case consists of two integers **n (0 < n <= 32), d (0 <= d <= 16)**. Input is terminated a test case where n=0 and d=0, you must not process this test case.

Output

For each test case, print three integers, n, d and the number of different strictly n-ary trees of level d, in a single line. There will be a single space in between two integers of a line. You can assume that you would not be asked about cases where you had to consider trees that may have more than 2^{10} nodes in a level of the tree. You may also find it useful to know that the answer for each test case will always fit in a 200 digit integer.

Example

```
Input:
2 0
2 1
```

```
2 2
2 3
3 5
0 0
```

Output:

```
2 0 1
2 1 1
2 2 3
2 3 21
3 5 58871587162270592645034001
```

QUESTION – 3

A set **S** of positive integers is called *strongly triple-free* if, for any integer **x**, the sets {**x**, 2**x**} and {**x**, 3**x**} are not subsets of **S**. Let's define **F(n)** as a number of strongly triple-free subsets of {1, 2, ..., **n**}, where **n** is a natural number.

You need to write a program which being given a number **n** calculates the number **F(n)** modulo 1 000 000 001.

Input

The first line of input contains integer **T** (1 ≤ **T** ≤ 500) - the number of testcases. Then descriptions of **T** testcases follow.

The description of the testcase consists of one line. The line contains an integer number **n** (1 ≤ **n** ≤ 100 000).

Output

For each testcase in the input your program should output one line. This line should contain one integer number which is the number **F(n)** modulo 1 000 000 001.

Example

Input:

```
5
3
1
10
20
39
```

Output:

```
5
2
```

```
198
43776
971827200
```

QUESTION – 4

Dear Uncle Jack is willing to give away some of his collectable CDs to his nephews. Among the titles you can find very rare albums of Hard Rock, Classical Music, Reggae and much more; each title is considered to be unique. Last week he was listening to one of his favorite songs, Nobody's fool, and realized that it would be prudent to be aware of the many ways he can give away the CDs among some of his nephews.

So far he has not made up his mind about the total amount of CDs and the number of nephews. Indeed, a given nephew may receive no CDs at all.

Please help dear Uncle Jack, given the total number of CDs and the number of nephews, to calculate the number of different ways to distribute the CDs among the nephews.

Input

The input consists of several test cases. Each test case is given in a single line of the input by, space separated, integers N (1 <= N <= 1000) and D (0 <= D <= 2500), corresponding to the number of nephews and the number of CDs respectively. The end of the test cases is indicated with N = D = 0.

Output

The output consists of several lines, one per test case, following the order given by the input. Each line has the number of all possible ways to distribute D CDs among N nephews.

Example

```
Input:
1 20
3 10
0 0
```

Output:
```
1
59049
```

QUESTION – 5

A fashion show rates participants according to their level of hotness. Two different fashion shows were organized, one for men and the other for women. A date for the third is yet to be decided ;) .

Now the results of both fashion shows are out. The participants of both the fashion shows have decided to date each other, but as usual they have difficuly in choosing their partners. The Maximum Match dating serive (MMDS) comes to their rescue and matches them in such a way that that maximizes the hotness bonds for all couples.

If a man has been rated at hotness level `x` and a women at hotness level `y`, the value of their hotness bond is `x*y`.

Both fashion shows contain **N** participants each. MMDS has done its job and your job is to find the sum of hotness bonds for all the couples that MMDS has proposed.

Input

The first line of the input contains an integer **t**, the number of test cases. **t** test cases follow.

Each test case consists of 3 lines:

- The first line contains a single integer **N** (1 <= **N** <= 1000).
- The second line contains **N** integers separated by single spaces denoting the hotness levels of the men.
- The third line contains **N** integers separated by single spaces denoting the hotness levels of the women.

All hotness ratings are on a scale of 0 to 10.

Output

For each test case output a single line containing a single integer denoting the sum of the hotness bonds for all pairs that MMDS has proposed.

Example

Input:
```
2
```

```
2
1 1
3 2
3
2 3 2
1 3 2
```

Output:

```
5
15
```