ACM ICPC

UM Qualification 2017

October 8th

International Collegiate Programming Contest

IBM event sponsor

Problems

A	Alien Permutation
B	Boxes
C	Cipher
D	DNA Sequence
E	Exam? No!
F	Falling Cookies
G	Gifts
H	Happy Girl
I	Identifying Crosses
J	Jump Trading
K	Knights of the Round Table
L	Legal Rook Board

This page is intentionally left blank.

Problem A
Alien Permutation

It is the year 3017. Humans have discovered many alien civilizations, and are no longer the only intelligent life in the known Universe. With invention of long-range teleportation, humans learned how to travel to the distant planets in different galaxies and became friends with lots of different alien species.

One day, one of your alien friends comes to you, and ask you a very simple question that appeared on his daughter's eight's grade homework: what is the next permutation of a given permutation?

As a computer scientist and, also, a mathematician, this problem looks super easy to you. However, you soon figure out that you friend's numbering system is not the same as yours, and this makes the problem much harder.

Thinking about this problem a little bit, trying to solve it, and also having messed up a lot of permutations, you decide to write a program to help you solve this for your friend.

Next permutation of a given permutation is the next permutation in lexicographical ordering. For example, in human system, next permutation of 2 1 3 4 is 2 1 4 3, and 4 3 2 1 is a maximal permutation.

Input

The input contains a single test case. The first line contains an integer N $(1 \leq N \leq 1000)$, which is the length of the permutation. The next line contains N different integers A_i $(1 \leq A_i \leq N)$, which represents number ordering in your alien friend's system. That is, $A_1 < A_2 < \cdots < A_N$. The next line contains N different integers B_i $(1 \leq B_i \leq N)$, and you need to calculate the next permutation of B in the number system A.

Output

A single line of the permutation. If the given permutation is already a maximal permutation, output "noitatumrep lamixam" instead. This means "maximal permutation" in your friend's language.

Sample Input 1

```
5
5 4 3 2 1
5 4 3 2 1
```

Sample Output 1

```
5 4 3 1 2
```

Sample Input 2

```
5
5 3 2 4 1
1 4 2 3 5
```

Sample Output 2

```
noitatumrep lamixam
```

Problem B
Boxes

Mr Apex is a distinguished professor of Computational Geometry. While preparing for his next class, he realized that he needs to draw three-dimensional pictures for his students. Mr Apex can only draw two-dimensional figures in two dimensions. Please help him to draw those three-dimensional pictures.

Apex has a board of size N by M in his classroom. At each position, he put a number A_{ij} indicating the number of boxes that are to be stacked on top of each other. He would like that to be converted to a 3-dimensional drawing, and projected onto a 2-dimensional space of a board.

A box is a three-dimensional object of the following format -- the corners are represented by a `+`, and the edges are represented by `-`, `/` or `|`. The face of the box is filled with spaces, and the background is represented by . (a dot).

For example, the following box has length of 3, width of 1 and height of 2:

```
..+---+
./   /|
+---+ | H = 2
|   | +
|   |/. W = 1
+---+..
L = 3
```

A three-dimensional picture follows the following rules, if two boxes are adjacent:

<table>
<tr><th>Left and right</th><th>Front and back</th><th>Up and down</th></tr>
<tr>
<td><pre>
..+---+---+
./   /   /|
+---+---+ |
| | | +
| | |/.
+---+---+..
</pre></td>
<td><pre>
....+---+
.../   /|
..+---+ |
./   /| +
+---+ |/.
| | +..
| |/...
+---+....
</pre></td>
<td><pre>
..+---+
./   /|
+---+ |
| | +
| |/|
+---+ |
| | +
| |/.
+---+..
</pre></td>
</tr>
</table>

For the output, your picture must cover the entire background. That is, none of the top, bottom, leftmost, or rightmost line should be filled completely by . . (dots)

Input

The first line contains five integers: N, M, L, W, H $(1 \le N, M \le 50, 1 \le L, W, H \le 10)$, as described in the problem statement. The next is a matrix of N lines, where each line contains M integers. The jth integer in the $i+1$th line represents A_{ij} $(1 \le A_{ij} \le 20)$. The first line of matrix is to be drawn to be behind the line that follows it. That is, the first line will end up being furthest in the background on the 3-d drawing, and the last line in the matrix will be on the foreground of the 3-d drawing.

All boxes are identical to each other and have the size as specified by L, W, H.

Output

The three-dimensional picture.

Sample Input

```
4 12 3 1 2
2 1 1 1 2 1 1 2 2 1 2 2
2 1 1 1 2 1 1 2 1 2 1 2
2 1 1 1 2 1 1 2 1 1 1 2
1 2 2 2 1 1 1 2 1 1 1 2
```

Sample Output

```
........+---+...........+---+.......+---+---+...+---+---+
......./   /|........../   /|....../   /   /|../   /   /|
......+---+ |.........+---+ |.....+---+---+---+---+---+ |
...../   /| +---+---+/   /| +---+/   /|  /   /|  /   /| +
....+---+ |/   /   /+---+ |/   /+---+ | +---+ | +---+ |/|
.../   /| +---+---+/   /| +---+/   /| +-|   | +/   /| + |
..+---+---+---+---+---+ |/   /+---+ |/  |   |/+---+ |/| +
..|  /   /   /   /|   | +---+/   /| +---+---+/   /| + |/.
..| +---+---+---+ |   |/   /+---+ |/   /   /+---+ |/| +..
..+-|   |   |   | +---+---+-|   | +---+---+-|   | + |/...
./  |   |   |   |/   /   /  |   |/   /   /  |   |/| +....
+---+---+---+---+---+---+---+---+---+---+---+---+ |/.....
|   |   |   |   |   |   |   |   |   |   |   |   | +......
|   |   |   |   |   |   |   |   |   |   |   |   |/.......
+---+---+---+---+---+---+---+---+---+---+---+---+........
```

Problem C
Cipher

Good morning, agent W-12. Your mission, should you choose to accept it, is as follows:

You have just intercepted an encrypted data stream from the enemy, and your task is to decrypt it. Decryption is generally very hard, but you know that the enemy uses a very simple technique to generate the cipher text from the plain text.

The transformation from one to the other (and back) can be represented by a one to one mapping between two permutations of the alphabet. Note that anything other than letter characters remain the same.

```
Plain:    ABCDEFGHIJKLMNOPQRSTUVWXYZ
Cipher:   MNOCDXABIJKLEFGPQRYZHSTUVW
```

It is still hard to decrypt since you only know the cipher alphabet is a permutation, but you don't know which. Thus you need to try all 26! ways in the worst case. However, you know exactly how many times each letter appears in the plain text! This makes the decryption process easier.

Input

The first line contains one integer: n $(1 \leq n \leq 26)$, which is the number of different letters in the plain text. The following n lines, each line contains a lower-case letter A and a number B, which represents that letter A appears B times in the plain text. Note that the letter count is case insensitive, but your output plain text must preserve cases. The final line of the input contains a string: the cipher text, which contains at most 5000 characters.

All inputs are valid, and it is guaranteed that for different A, the number B is different.

Output

A single string: the plain text.

Sample Input

```
10
a 1
b 2
e 3
l 4
m 5
o 6
p 7
r 8
s 9
y 10
Qjay pfchwqg: hqwwwgGGgccccccpPPpppfFffFffaAaaaAaayyYyyyyyy.
```

Sample Output

```
Easy problem: bellllmMMmoooooopPPpppprRrrRrrsSsssSssyyYyyyyyy.
```

Problem D
DNA Sequence

What is DNA?

Deoxyribonucleic acid (DNA) is a molecule that carries the genetic instructions used in the growth, development, functioning and reproduction of all known living organisms and many viruses. DNA and RNA are nucleic acids; alongside proteins, lipids and complex carbohydrates (polysaccharides), they are one of the four major types of macromolecules that are essential for all known forms of life. Most DNA molecules consist of two biopolymer strands coiled around each other to form a double helix.

The two DNA strands are termed polynucleotides since they are composed of simpler monomer units called nucleotides. Each nucleotide is composed of one of four nitrogen-containing nucleobases—either cytosine (C), guanine (G), adenine (A), or thymine (T)—and a sugar called deoxyribose and a phosphate group.

A group of scientists have just successfully separated a single strand of DNA sequence. After purification, anything other than nucleobases is gone. However, the purification process destroys the original strand, and now the scientists only know the number of cytosine, guanine, adenine and thymine respectively in the original strand.

The Problem:

The scientists would like to know what is the expected position of the first cytosine (the C of the C, G¸ A, T) in the original strand if all possible strands are equally likely. For simplicity, assume that all sequences are possible, regardless of the biological laws. Note that if there are N nucleobases where nucleobase is one or more of C, G, A, T, the number of possible sequences is $N!$. This holds, even if all the nucleobases are the same.

Input

The first line of the input contains an integer N $(1 \leq N \leq 100)$, the number of test cases. The next N lines, each line contains four integers, C, G, A, T $(1 \leq C \leq 10^9, 0 \leq G, A, T \leq 10^9)$, which represents the number of cytosine, guanine, adenine and thymine, respectively.

Output

For each test case, display its case number followed by the expected position in irreducible fractions. (It can be easily proven that the answer is a rational number.) The positions start at 1. Follow the format of the sample output and pay attention to the spaces.

Sample Input

```
2
1 0 0 0
1 1 1 1
```

Sample Output

```
Case 1: 1 / 1
Case 2: 5 / 2
```

Hint

Note: In case 2, there are 4! different possible strands. Among all of them, the first cytosine (C) is at the 1st, 2nd, 3rd, 4th place the same number of times: 6 times. Thus the expected position is $(1+2+3+4)\times 6 \div 24 = 2.5$ = 5 / 2.

Problem E
Exam? No!

No one likes final exams. Being aware of this, the professor decides to give all the students in his class a chance to get rid of the final exam.

There are $2n$ students in the class, and all of their names are different. The professor puts a row of $2n$ boxes in a room. Each box represents a single student, and there is a nametag outside each box. Inside each box, there is another nametag, which may be or may not be the same as the name outside the box. The students enter the room one at a time, and each student can open at most n boxes, one at a time. If the student did not see his or her own name, then he or she has to take the exam!

The students may have a discussion before the first student enters the room with the boxes, but after that, there is no further communication. Note that students don't have to decide which boxes to open before they enter the room. A student may not leave a message of any kind for other students. In particular, all the boxes are shut once a student leaves the room.

Assume that all the students are very smart, and an optimal strategy is used. What is the probability that there is no final for the entire class? In other words, what is the probability that no one needs to take the final exam? Note that no student is selfish, and each of them are trying to maximize the probability that no one needs to take the exam, regardless of their own chance.

Input

The input contains several test cases. The first line contains a number T $(1 \leq T \leq 10)$, which represents the number of test cases. In the following T lines, each line contains an integer n $(1 \leq n \leq 1000)$.

Output

For each test case, display its case number followed by the maximum probability required. Your output should have exactly six digits to the right of the decimal point. Follow the format of the sample output.

Sample Input

```
2
1
10
```

Sample Output

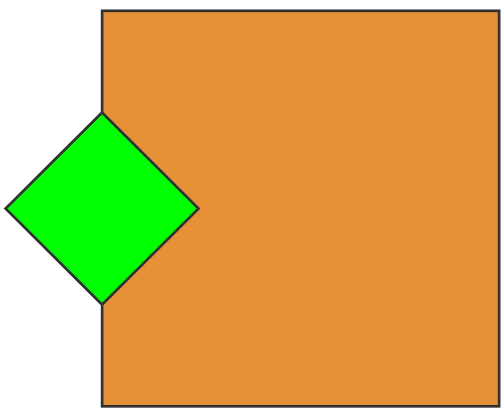
```
Case 1: 0.500000
Case 2: 0.331229
```

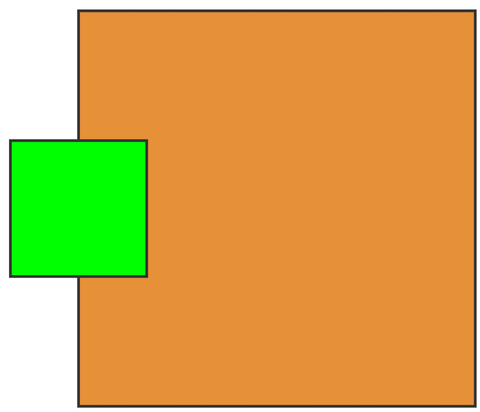
Problem F
Falling Cookies

Frank and his friends are playing a game. They have a lot of cookies of different shapes, which you can assume are a simple polygon, though not necessarily convex. For each cookie, Frank and his friends are trying to place the cookie at the boundary of a table, so that the outmost point of the cookie is as far away from the table's edge as possible. This is easy if you could put a cookie anywhere, since you can put it completely outside of the table. However, that way, the cookie will fall. For some cookies, Frank doesn't want this to happen, since he is going to eat them after the game. For other cookies, Frank doesn't like them, and wants to secretly destroy them by pretending to lose the game.

Please help Frank to find a way to put the cookies such that:

1. the distance of the outmost point from the table is maximized, and the cookie is not falling;
2. the distance of the outmost point from the table is minimized, and the cookie is falling.





You are allowed to rotate cookies in whatever way you want. Note that the boundary of a table is a straight line, and you cannot put the cookies at the corner of the table.

Input

The input consists of a single test case. The test case starts with a single integer n ($3 \leq n \leq 100$), the number of points of the polygon used to describe the cookie's shape. The following n pairs of integers x_i, y_i ($-2\,000 \leq x_i \leq 2\,000, 0 \leq y_i \leq 2\,000$) are the coordinates of the polygon points in order, either clockwise or counter clockwise.

Output

The output contains two lines. The first line contains a single number that represents the maximum distance outside the table such that the cookie will not fall (answer for query 1). The second line contains a single number that represents the minimum distance outside the table such that the cookie will fall (answer for query 2). Your output should have exactly six digits to the right of the decimal point.

Sample Input

```
4
0 0
0 10
10 10
10 0
```

Sample Output

```
7.071068
5.000000
```

Problem G
Gifts

Richard receives a lot of gifts for his birthday. There are a lot of them, but all the gifts can be classified into three categories: A, C, or M.

Richard wants to use a machine to help him classify all the gifts. N friends of Richard form a line, and each of them holds up a gift, which is either an A, a C, or an M. They give their gifts to the machine one by one sequentially.

At each moment, the machine can decide whether to open the warehouse, choose one of the categories, and put all the gifts of that category that it currently holds into the warehouse, and then close the ware house. Also, the machine can hold at most M gifts at a time. Richard wants the machine to minimize the number of access to the warehouse.

Here is an example. Let $N = 6$ and $M = 2$, and the gift sequence be ACMACM.

1. The machine gets an A.
2. The machine gets a C.
3. The machine has to open the warehouse, since it can hold at most 2 gifts. Then it chooses to put the C into the warehouse.
4. The machine gets an M.
5. It puts M into the warehouse.
6. It gets an A.
7. It puts two A's into the warehouse.
8. It gets a C.
9. It gets an M.
10. It puts C into the warehouse.
11. It puts M into the warehouse.

The total number of access to the warehouse is 5. (Steps 2, 5, 7, 10, 11, where in line 7 both A's were put into the warehouse.)

Input

The first line contains two integers: N, M $(1 \leq N \leq 1000, 1 \leq M \leq 100)$.
The following line contains a string of length N. The string will only contain A, C and M.

Output

A single integer: the minimum number of access to the warehouse.

Sample Input

```
6 2
ACMACM
```

Sample Output

```
5
```

Problem H
Happy Girl

Alice, has been dreaming of being a movie star for a long long time. Her time is now, for several filmmaking companies invited her to play the main role in their own films. She is so happy!

Unfortunately, all these companies will start making their movies at the same time, and the greedy Alice doesn't want to miss any of them!

You are asked to tell her whether she can act in all the films, or not. As for a film,

1. it will be made ONLY on some fixed days in a week, i.e., Alice can only work for the film on these days;
2. Alice MUST work on each film for at least a specified number of days;
3. the film MUST be finished before a prearranged deadline.

For example, assuming a film work can be schedule only on Monday, Wednesday and Saturday; and Alice can work for the film for 4 days; and it must be finished within 3 weeks. In this case where she can work for the film on Monday of the first week, on Monday and Saturday of the second week, and on Monday of the third week.

Notice that on a single day Alice can work on at most ONE film.

Input

The first line of the input contains a single integer T $(1 \leq T \leq 20)$, the number of test cases. Then T cases follow. Each test case begins with a single line containing an integer N $(1 \leq N \leq 20)$, the number of films. Each of the following n lines is in the form of "F1 F2 F3 F4 F5 F6 F7 D W". Fi $(1 \leq i \leq 7)$ is 1 or 0, representing whether the film can be made on the i-th day in a week (a week starts on Sunday): 1 means that the film can be made on this day, while 0 means the opposite. Both D $(1 \leq D \leq 50)$ and W $(1 \leq W \leq 50)$ are integers, and Alice should be on the film for at least D days and the film must be finished in W weeks.

Output

For each test case print a single line, 'Yes' if Alice can attend all the films, otherwise 'No'.

Sample Input

```
2
2
0 1 0 1 0 1 0 9 3
0 1 1 1 0 0 0 6 4
2
0 1 0 1 0 1 0 9 4
0 1 1 1 0 0 0 6 2
```

Sample Output

```
Case 1: Yes
Case 2: No
```

Hint

A proper schedule for the first test case:

```
date     Sun    Mon    Tue    Wed    Thu    Fri    Sat
week1         film1  film2  film1         film1
week2         film1  film2  film1         film1
week3         film1  film2  film1         film1
week4         film2  film2  film2
```

Problem I
Identifying Crosses

Given an image, your goal is to determine whether the input image contains a single cross.

Here are some examples of a cross.

..*.. ..*.. ***** ..*.. ..*..	.*... ***** .*... .*...	.*.. ***. .*.. .*.. .*..

Here are some examples of NOT a cross.

..*.. ..*.. ***.. ..*.. ..*..	..... ***** .*... .*...	.*.. ***. .*.. .*.. .*.*

That is to say, for a cross, it needs to first have a center, and then all four directions must be non-empty. Also, all pixels on the image other than the cross must be dots.

Input

Input will consist of multiple input sets. Each set will start with a line with two integers: n, m $(1 \leq n, m \leq 50)$, the size of the image. The next n lines represent the image, in the same way as shown in the problem statement. There will be only * or . in the image. The line 0 0 will indicate end of input and should not be processed. There will be no more than 100 input sets in one case.

Output

For each input set, output a single line showing that whether the input image contains a single cross. Follow the format of the sample output.

Sample Input

```
5 5
.*...
*****
.*...
.*...
.....
5 4
.*..
***.
.*..
.*..
.*.*
0 0
```

Sample Output

```
Case 1: Yes
Case 2: No
```

Problem J
Jump Trading

Collin is traveling in country J. In country J, there are N big cities and M roads, and each road connects two different cities. Some roads are undirected, while others are directed. The country J is very big, which results in differences in prices for the same goods in different cities. Noting this fact, Collin wants to make some money while traveling in country J. The merchandise he chooses is a crystal ball.

Collin can visit each road or city as many times as he wants, but he must start at city 1, and ends his journey at city N. Also, the main purpose for his journey is traveling, and thus he wants to do the trading part at most once. That is, he is going to buy a crystal ball at some city A, and then sell it at some city B. If Collin cannot make any money by doing any trading, he doesn't have to do it.

Input

The first line contains two integers, N $(1 \leq N \leq 50000)$ and M $(1 \leq M \leq 100000)$, which represents the number of cities, and the number of roads. The second line contains N integers. The i-th integer represents the price of a crystal ball in city i. All prices are between 1 and 100.

The following M lines, each line contains three integers a, b, c $(1 \leq a, b \leq N, 1 \leq c \leq 2)$, represents a road from city a to city b. If $c = 1$, then the road is directed, and Collin can only go from a to b. If $c = 2$, then the road is undirected, and Collin can go in either direction.

Output

A single integer represents the maximum money Collin can earn while traveling. If he cannot make any money, output 0.

Sample Input

```
5 5
4 3 5 6 1
1 2 1
1 4 1
2 3 2
3 5 1
4 5 2
```

Sample Output

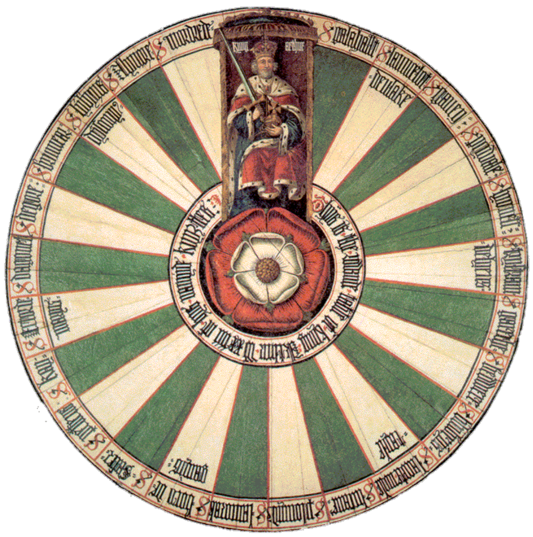
```
5
```

Explanation

Here Collin travels from city 1, to 2, 3, 5, 4. At city 4 he buys the crystal ball costing him 6 monetary units. He then travels to city 5, and sells the ball for 1 monetary unit. His total profit is 6 – 1 = 5.

Problem K
Knights of the Round Table

The Round Table is King Arthur's famed table in the Arthurian legend, around which he and his Knights congregate. As its name suggests, it has no head, implying that everyone who sits there has equal status. Though everyone is equal, King Arthur still has a hard time to decide the place for each knight.



As you may have noticed, King Arthur does not have enough time to assign seats for knight, and thus it is your job now. Before that, he'd like to know how many different possible seat assignments are there. Two assignments can be seen equal if they can be transformed to each other by rotation of the table.

Input

The first line contains an integer T $(1 \le T \le 10)$, which represents the number of test cases. The following T lines, and each line contains an integer n, the total number of people around the table, including King Arthur.

In this problem, $1 \le n \le 10^{18}$. Yes, you are right, King Arthur is very powerful, and he can get that many knights to sit around his table. But he can also be alone, and feel lonely.

Output

For each test case, display its case number followed by the number of ways to assign seats. Note that the number could be very big, thus instead of the exact number, output the number mod 1000000007. Follow the format of the sample output.

Sample Input

```
2
1
10
```

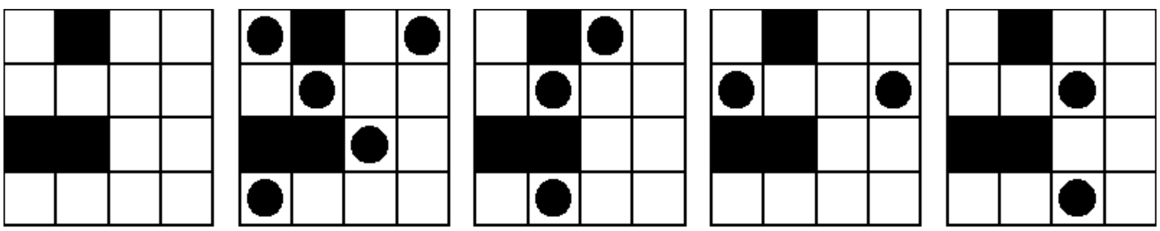
Sample Output

```
Case 1: 1
Case 2: 362880
```

Problem L
Legal Rook Board

In chess, the rook is a piece that can move any number of squares vertically or horizontally. In this problem we will consider small chessboards (at most 4×4) that can also contain walls through which rooks cannot move. The goal is to place as many rooks on a board as possible so that no two can capture each other. A configuration of rooks is legal provided that no two rooks are on the same horizontal row or vertical column unless there is at least one wall separating them.

The following image shows five pictures of the same board. The first picture is the empty board, the second and third pictures show legal configurations, and the fourth and fifth pictures show illegal configurations. For this board, the maximum number of rooks in a legal configuration is 5; the second picture shows one way to do it, but there are several other ways.



Your task is to write a program that, given a description of a board, calculates the maximum number of rooks that can be placed on the board in a legal configuration.

Input

The input file contains one or more board descriptions, followed by a line containing the number '0' that signals the end of the file. Each board description begins with a line containing a positive integer n that is the size of the board; n, as mentioned, will be at most 4. The next n lines each describe one row of the board, with a '.' indicating an open space and an uppercase 'X' indicating a wall. There are no spaces in the input file. There will be no more than 100 cases in each test.

Output

For each test case, output one line containing the maximum number of rooks that can be placed on the board in a legal configuration. Follow the format of the sample output.

Sample Input

```
4
.X..
....
XX..
....
2
XX
.X
3
.X.
X.X
.X.
3
...
.XX
.XX
4
....
....
....
....
0
```

Sample Output

```
Case 1: 5
Case 2: 1
Case 3: 5
Case 4: 2
Case 5: 4
```