The 2021 icpc
Asia Nanjing Regional Contest

Practice Session

December 3, 2021

Problem List

A	Kangaroo Puzzle
B	Ah, It's Yesterday Once More
C	Computer Science Ability Test
D	Nihongo wa Muzukashii Desu

This problem set should contain 4 (four) problems on 6 (six) numbered pages. Please inform a runner immediately if something is missing from your problem set.

Prepared by SUA Programming Contest Problem Setter Team.
https://sua.ac/

Hosted by

Problem Set Prepared by

Problem A. Kangaroo Puzzle

Your friend has made a computer video game called "Kangaroo Puzzle" and wants you to give it a try for him. As the name of this game indicates, there are some (at least 2) kangaroos stranded in a puzzle and the player's goal is to control them to gather. As long as all the kangaroos in the puzzle get together, they can escape the puzzle by the miraculous power of kangaroos.

The puzzle is a $n \times m$ grid consisting of nm cells. There are walls in some cells and the kangaroos cannot enter these cells. The other cells are empty. The kangaroos can move in the following direction: up, down, left and right. It is guaranteed that one kangaroo can move from an empty cell to any other. It is also guaranteed that there is no cycle in the puzzle — that is, it's impossible that one kangaroo can move from an empty cell, pass by several distinct empty cells, and then back to the original cell.

There is exactly one kangaroo in every empty cell at the beginning. You can control the kangaroos by pressing the button `U`, `D`, `L`, `R` on your keyboard. The kangaroos will move simultaneously according to the button you press. For instance, if you press the button `U`, a kangaroo would move to the upper cell if it exists and is empty; otherwise, the kangaroo will stay still. You can press the buttons for at most 50000 times. If there are still two kangaroos standing in different cells after 50000 steps, you will lose the game.

Input

The first line contains two integers, n and m ($1 \leq n, m \leq 20$), the height and the width of the puzzle, respectively. Each of the next n lines contains a (`0,1`)-string of length m, representing the puzzle. If the j-th character of the $i+1$-th line is `1`, then the cell at the i-th row and the j-th column is empty; otherwise (i.e. it is `0`), the corresponding cell is blocked and cannot be entered.

Output

Print a string consisting of `U`, `D`, `L`, `R`, such that all kangaroos will get together after pressing the buttons in the order of this string. The length of the string should not exceed 50000. There are many possible valid answers, so just print any of them.

Examples

standard input	standard output
4 4 1111 1001 1001 1110	LLUUURRRDD
2 15 111111111111111 101010101010101	ULLLLLLLLLLLLLL

Problem B. Ah, It's Yesterday Once More

In 2018, hosted by Nanjing University of Aeronautics and Astronautics (NUAA), the *International Collegiate Programming Contest* (ICPC) regional was held in Nanjing again after a few years' gap. There were over 400 teams in the contest and team *Power of Two* from Tsinghua University won the champion.

Two years have passed and after the great success in 2018 and 2019, NUAA continues to hold the ICPC Nanjing Regional in 2020. Although we can't gather in Nanjing this time due to the pandemic, we should still be grateful for the hard work done by all staff and volunteers for this contest. Thank you all for your great contribution to this contest!

The 2018 ICPC Asia Nanjing Regional Contest

In the 2018 contest, problem K, *Kangaroo Puzzle*, requires the contestants to construct an operation sequence for the game. Let's first recall the content of that problem:

> The puzzle is a grid with n rows and m columns ($1 \le n, m \le 20$) and there are some (at least 2) kangaroos standing in the puzzle. The player's goal is to control them to get together. There are some walls in some cells and the kangaroos cannot enter the cells with walls. The other cells are empty. The kangaroos can move from an empty cell to an adjacent empty cell in four directions: up, down, left, and right. It's guaranteed that kangaroos can reach from any empty cell to any other empty cells by going through adjacent empty cells. It is also guaranteed that there is no cycle in the puzzle – that is, it's impossible that one kangaroo can move from an empty cell, pass by several distinct empty cells, and then back to the original cell.
>
> There is exactly one kangaroo in every empty cell in the beginning and the player can control the kangaroos by pressing the button U, D, L, R on the keyboard. The kangaroos will move simultaneously according to the button you press. For instance, if you press the button R, a kangaroo would move one cell to the right if it exists and is empty, and will stay still if it does not exist or is not empty.
>
> In this problem, the contestant needs to construct an operating sequence of at most 5×10^4 steps consisting of U, D, L, R only. If after operating these steps in order there are still two kangaroos standing in different cells, the contestant will be given a "Wrong Answer" verdict.

Our dear friend, Kotori, also took part in the contest and submitted a code of randomized algorithm. To her surprise, this simple solution is judged as a correct answer. We now present her solution as follows:

```
#include <bits/stdc++.h>
char s[5] = 'UDLR';
using namespace std;
int main()
{
  srand(time(NULL));
  for (int i = 1; i <= 50000; i++) putchar(s[rand() % 4]);
  return 0;
}
```

For contestants who are not familiar with C and C++: the above code will output a random string of length 5×10^4 consisting only of characters 'U', 'D', 'L' and 'R', where each character has equal probability to appear in each position in the string.

Kotori suspects that things might not be that simple for this problem, so right now, in this *2020 ICPC Nanjing Regional* contest, you need to construct an input data to hack her solution. Due to the randomness, your input data only needs to satisfy a successful hacking rate of at least 25%.

Formally speaking, we've prepared 500 randomly generated strings that each character has equal probability to appear in each position and will use them as the controlling sequence against your answer. For your answer to be accepted, there should be at least 125 times that after using your answer as the map of cells and the whole controlling sequence is executed, there are still kangaroos in different cells.

Note that your input data should be completely legal. That is to say,

- The map in your answer should not be larger than 20×20;
- Your answer should contain at least two empty cells;
- All empty cells in your answer should be reachable starting from any empty cell;
- No cycles consisting of empty cells are allowed.

Input

There is no input for this problem. You're on your own!

Output

You should first output one line containing two integers n and m $(1 \le n, m \le 20)$ separated by a space, indicating the number of rows and columns of the map in your answer.

You should then output n lines where the i-th line contains a binary string $s_{i,1}s_{i,2}\cdots s_{i,m}$ ($s_{i,j} \in \{\text{'0'}, \text{'1'}\}$) of length m. If $s_{i,j} = \text{'1'}$ then the cell in the i-th row and the j-th column is empty; Otherwise that corresponding cell contains a wall and cannot be entered.

Note again that your answer only need to achieve a successful hacking rate of at least 25%. Not that hard isn't it?

Example

standard input	standard output
(No input)	3 4 1111 1010 1100

Note

The sample output we provide you is (obviously) incorrect. It only serves the purpose of showing you the output format. This is a 3×4 map with 4 walls, so there will be 8 kangaroos in the empty cells at the beginning.

Problem C. Computer Science Ability Test

Computer Science Ability Test (CSAT) aims to evaluate objectively, through unified examinations with 8 true or false questions, the abilities of testees in computer science.

Kotori is taking the test, but soon she discovers with dismay that she cannot answer even a single question. Now she presents you with these 8 questions. Can you tell her the correct answer to each question?

1. (On Um_nik) If you know at least 3 of these things and you are not red. You are doing it wrong. Stop learning useless algorithms, go and solve some problems, learn how to use binary search.
2. (On Dota2) PSG.LGD beat Team Spirit in The International 10 and won the Champion.
3. (On Complexity) It is NP-hard to approximate the minimum consistent DFA problem to within a factor of $n^{1-\epsilon}$.
4. (On Data Structure) Because AVL tree is strictly balanced, so for each node, if the balance factor changes, this node needs rotation adjustment.
5. (On Linear Systems) Solving a general linear system $\mathbf{A}\boldsymbol{x} = \boldsymbol{b}$ over $\mathbb{Z}_p$ can be reduced to solving a unit-weight Laplacian system $\hat{\mathbf{L}}$ of size $O\left(\mathrm{nnz}(\mathbf{A}) \log^2 p / \log\log p\right)$.
6. (On Theory of Computation) Let $L(M)$ be the language that the Turing machine M accepts, then the language $\{``M"|M$ is a Turing machine and $L(M)$ is uncountable$\}$ is recursively enumerable but not recursive.
7. (On Math) $\cos(\cos(\cos(x))) > \sin(\sin(\sin(x)))$ holds for all $x \in \mathbb{R}$.
8. (On Physics) One mole of ideal gas is expanded from V_0 to $2V_0$ in a reversible adiabatic process. If the temperature of the gas decreases by 25%, the gas may be a type of monatomic gas.

Input

This problem has no input. You're on your own!

Output

Output one line containing 8 characters. The characters must be either 'T' or 'F', where 'T' means true and 'F' means false. The i-th character indicates the answer to the i-th true or false question.

Example

standard input	standard output
(No input)	FTFTFTFT

Note

Note that the sample output is not the correct answer! It only serves the purpose of showing you the output format.

Problem D. Nihongo wa Muzukashii Desu

Japanese is one of the most difficult languages to learn in the world. Among all those twisted grammar rules, the most troublesome ones for the beginners must be the *verb conjugation* rules.

Japanese verbs appear in different forms under different contexts. By the conjugation rules between their different forms, Japanese verbs can be roughly grouped into three types. We now introduce you the *masu form* to *te form* conjugation rule for the first type of verbs.

- We say a verb is in *masu form* if it ends with "masu" (ます). For example, "naraimasu" (習います, learn) and "nomimasu" (飲みます, drink) are all masu form verbs.
- We say a verb is in *te form* if it ends with "te" (て) or "de" (で). For example, "naratte" (習って, learn) and "nonde" (飲んで, drink) are all te form verbs.
- If the masu form of a verb ends with "imasu" (います), "chimasu" (ちます) or "rimasu" (ります), to change it into its te form, we remove the "imasu", "chimasu" or "rimasu" at the end and append "tte" (って) to it. For example, "kaimasu" (買います, buy) → "katte" (買って), "machimasu" (待ちます, wait) → "matte" (待って) and "kaerimasu" (帰ります, return) → "kaette" (帰って).
- If the masu form of a verb ends with "mimasu" (みます), "bimasu" (びます) or "nimasu" (にます), to change it into its te form, we remove the "mimasu", "bimasu" or "nimasu" at the end and append "nde" (んで) to it. For example, "nomimasu" (飲みます, drink) → "nonde" (飲んで), "yobimasu" (呼びます, call) → "yonde" (呼んで) and "shinimasu" (死にます, die) → "shinde" (死んで).
- If the masu form of a verb ends with "kimasu" (きます), to change it into its te form, we remove the "kimasu" at the end append "ite" (いて) to it. For example, "kakimasu" (書きます, write) → "kaite" (書いて). But there is only one verb this rule does not apply, which is the verb "ikimasu" (行きます, go) → "itte" (行って).
- If the masu form of a verb ends with "gimasu" (ぎます), to change it into its te form, we remove the "gimasu" at the end and append "ide" (いで) to it. For example, "isogimasu" (急ぎます, hurry) → "isoide" (急いで).
- If the masu form of a verb ends with "shimasu" (します), to change it into its te form, we remove the "shimasu" at the end and append "shite" (して) to it. For example, "kashimasu" (貸します, lend) → "kashite" (貸して).

It's time to check how much you've learnt in this lesson! Given a Japanese verb of the first type in its masu form represented in romaji (which means in lower-cased English letters), please change it into its te form.

You might have noticed that if we represent a Japanese verb in romaji, for example "nomimasu", it's hard to tell whether this verb ends with "imasu" or "mimasu" for the beginners (actually it ends with "mimasu" as "mi" is one syllable). To simplify this problem, we will not provide you with verbs ending with "imasu".

Input

There are multiple test cases. The first line of the input contains an integer T (about 100) indicating the number of test cases. For each test case:

The first and only line contains a string s ($1 \le |s| \le 30$) which is a Japanese verb of the first type in its masu form presented in romaji. This verb is guaranteed to end with "chimasu", "rimasu", "mimasu", "bimasu", "nimasu", "kimasu", "gimasu" or "shimasu".

Output

For each test case output one line containing one string indicating the te form of the verb in romaji.

Example

standard input	standard output
10 machimasu kaerimasu nomimasu yobimasu shinimasu kakimasu ikimasu kikimasu isogimasu kashimasu	matte kaette nonde yonde shinde kaite itte kiite isoide kashite