

# Day 1 Question 1: Subsets

**Your solution:** N:\subset\subset.{pas,c,cpp}

**Input file:** subset.in

**Output file:** subset.out

In this problem, you will write a program to find the minimal solution to a set of *set inequalities*. A set inequality has the format

$X$  contains  $S$

where  $X$  may be any set name and  $S$  may be a set name or set element. If  $S$  is a set name the inequality means that  $X$  is a superset or equal to  $S$ . If  $S$  is an element the inequality means that  $X$  contains  $S$ . Sets are named A-Z and contain elements from a-z.

The first line of input specifies the number of set inequalities ( $M$ ). The next  $M$  lines each contain one set inequality. For each set name that appears in the input, your program must determine its minimal set: the smallest set of elements that the name must take in order that all of the inequalities hold. Output, in alphabetical order, each set name followed its minimal set, with the elements in alphabetical order, in the format shown below.

## Sample Input

```
9
A contains B
A contains c
B contains d
F contains A
F contains z
X contains Y
Y contains X
X contains x
Q contains R
```

## Output for Sample Input

```
A = {c, d}
B = {d}
F = {c, d, z}
Q = {}
R = {}
X = {x}
Y = {x}
```

# Day 1 Question 2: Ransom Note

**Your solution:** N:\ransom\ransom.{pas,c,cpp}

**Input file:** ransom.in

**Output file:** ransom.out

Gilbert Bates, the magnate of aluminum siding, doors, and windows, has been kidnapped. You are to help the perpetrators produce a ransom note. Your raw materials are the text of a newspaper and the text of the ransom note. The ransom note is to be produced by clipping letters or strings of letters (and possibly spaces) from the newspaper and pasting them onto a blank sheet of paper to form the note. Your job is to determine the minimum number of pieces of paper that must be clipped and pasted to form the note. Between each pair of words in the note, either the clipping must contain a space or a boundary between clippings must occur (so that the blank background shows through).

## Input

The input file consists of the text of the note followed by the text of the newspaper. The text of the note is a single line, less than 2KB in length, in lowercase with no punctuation. The text of the newspaper is in upper and lower case with punctuation and newlines mixed in. Case may be ignored (aS IN aNY stANDARD RANSOM nOTE) and punctuation and newlines should not be clipped. The kidnappers have acquired a large number of copies of the same newspaper, so that the same or overlapping text may be clipped as many times as necessary. Every letter of the alphabet occurs at least once in the newspaper. The newspaper is smaller than 9KB in length. In at least 60% of the test cases the newspaper will be smaller than 2KB. An inefficient algorithm will probably exceed the time limit (about 15 seconds) when solving the larger cases.

## Output

Print the minimum number of clippings followed by the clippings, one per line, in the correct order to compose the note. The case of the newspaper text should be preserved.

## Sample Input

```
drop the price on new thermopanes now or else
Rain Users Guide (by J.Z.)
While "rain" was intended to be a general purpose tool, at the time of
writing the primary goal was to study one particular software system.
As a result, some steps that are only done once (such as extracting
information from the program under study) are done using cumbersome
ad-hoc techniques that require significant manual intervention. While
"rain" can be used on arbitrary programs, more development work needs
to be done before this is a convenient process.
```

## Possible Output for Sample Input

```
19
d
ro
p
the pri
ce
```

on  
ne  
w  
the  
rm  
op  
an  
es  
n  
o  
W  
or  
el  
se

## Judging Input

To help you test your program, a very large input file **N:\ransom\ransom1.in** has been given to you. This will be one of the files used by the judges to evaluate your solution.

## Day 1 Question 3: The Game of 31

**Your solution:** N:\game\game.{pas,c,cpp}

**Input file:** game.in

**Output file:** game.out

The game of 31 was a favourite of con artists who rode the railroads in days of yore. The game is played with a deck of 24 cards: four labelled each of 1, 2, 3, 4, 5, 6. (That is, there are four cards labelled '1', four cards labelled '2', and so on.) Initially all of the cards are spread, face up, on a table and the "discard pile" is empty. The players then take turns. During each turn, a player picks up one unused card from the table and lays it on the discard pile. The object of the game is to be the last player to lay a card such that the sum of the cards in the pile does not exceed 31. Your task is to determine the eventual winner of a partially played game, assuming each player plays the remainder of the game using a perfect strategy.

For example, in the following game player B wins:

```
Player A plays 3
Player B plays 5
Player A plays 6
Player B plays 6
Player A plays 5
Player B plays 6
```

### Input

The first line of the input file is the number of test cases. It is followed by one line for each test case. Each such line consists of a sequence of zero or more digits representing a partially completed game. The first digit is player A's move; the second player B's move; and so on. You are to complete the game using a perfect strategy for both players and to determine who wins.

### Output

For each game, output A or B on a single line to indicate the eventual winner of the game.

### Sample Input

```
5
356656
35665
3566
111126666
552525
```

### Output for Sample Input

```
B
B
A
A
A
```