Frisco FirstBytes Programming Contest

October 15, 2016

ADVANCED PROBLEM SET

DO NOT OPEN THE PACKET UNTIL TOLD TO DO SO.

The final <u>Advanced</u> winner will be decided ONLY by the number of <u>Advanced</u> problems solved. *Advanced Teams will receive NO credit for solving Beginner or Novice Problems!*

A Got Change?

Data File: A.txt
Time allowed = 10 seconds

Given a denomination coin set, such as $\{1, 5, 10, 25, 50, 100\}$, what is the smallest number of coins needed to make up 87 cents? There is a greedy algorithm for this;

> Use the largest denomination as many times as possible, then repeat the process with each lower denomination until the required amount is reached.

For 87c the minimum number of coins is five:

$$0\text{x}100\text{c} + 1\text{x}50\text{c} + 1\text{x}25\text{c} + 1\text{x}10\text{c} + 0\text{x}5\text{c} + 2\text{x}1\text{c}$$

But if the denomination set is $\{1, 5, 10, 12, 25, 50, 100\}$, the greedy method will not always give the best answer. Try 16c for example.

Given an amount to be changed and a denomination set, find the minimum number of coins needed to make up the given amount.

Input:
The input file begins with integer $T < 100$, on a line by itself, giving the number of test cases. Each test case follows on a single line and comprises space-separated integers. The amount, A, $A < 1000$, to be changed comes first, followed by the denomination set, given as a series of distinct integers in increasing order of size. The largest denomination will not be larger than 1000.

Output:
For each test case print one line containing the minimum number of coins needed to make up the given amount.

Sample Input	Sample Output
3	
87 1 5 10 25 50 100	5
16 1 5 10 25 50	3
16 1 5 10 12 25	3

B Don't Get Left With the Pepper

Data File = B.txt
Time limit = 10 seconds

Two players, Joe and Jane, play a simple game. They are presented with a large jar containing N wrapped pieces of candy and one chili pepper. They take turns, each removing one, two, or three pieces of candy from the jar. The object of the game is to leave your opponent with the chili pepper which, of course, he/she must eat.

Joe and Jane are both expert players of this game.

Given a game situation and whose turn it is to remove some candy, figure out who will win.

Input:
The input file will begin with integer T, the number of test cases, $T < 20$. Each test cases comprises one line of text containing N, the number of pieces of candy left in the jar, and the name of the player that is to remove candy next.

Output:
For each test case output one of the lines, "Joe wins" or "Jane wins".

Sample Input	Sample Output
2	
53 Joe	Joe wins
24 Jane	Joe wins

C Postal Delivery Route

Input File: C.txt
Run Time Limit: 10 sec

Given a street map with no dead-ends, cul-de-sacs or one-way-streets, is there a route that requires the postal delivery person to traverse every street exactly once?

There are houses on only one side of each street and there is a path via the streets from any intersection to any other intersection.

Your route must begin and end at the same intersection.

Input:
The input will begin with a line containing N ($1 \leq N \leq 20$), the number of test cases. Each test case begins with a line containing two integers M ($1 \leq M \leq 20$) and P ($1 \leq P \leq 50$), where there are M intersections, numbered $1, 2, \cdots, M$, and P streets. P lines follow, each one containing two integers, giving the intersection numbers at the ends of that street.

Output:
For each test case, print the the word "yes" or "no" according to whether there is a route that meets the above requirements.

Sample Input	Expected Output
3 4 4 1 2 2 3 3 4 4 1 4 5 1 2 2 3 3 4 4 1 1 3 5 6 2 1 3 1 4 5 2 3 2 4 5 3	yes no no

D Alien Sort

Data File: D.txt
Time allowed = 10 seconds

We know the normal alphabetical order of the English alphabet, and we can then sort words or other letter sequences. For instance these words are sorted:

ANTLER
ANY
COW
HILL
HOW
HOWEVER
WHATEVER
ZONE

The standard rules for sorting letter sequences are used:

- The first letters are in alphabetical order.
- Among strings with the same prefix, like the prefix AN in ANTLER and ANY, they are ordered by the first character that is different, T or Y here.
- One whole string may be a prefix of another string, like HOW and HOWEVER. In this case the longer sequence comes after the shorter one.

The Gorellians, at the far end of our galaxy, have discovered various samples of English text from our electronic transmissions, but they did not find the order of our alphabet. Being a very organized and orderly species, they want to have a way of ordering words, even in the strange symbols of English. Hence they must determine their own order. Unfortunately they cannot agree, and every Gorellian year, they argue and settle on a new order.

For instance, if they agree on the alphabetical order

UVWXYZNOPQRSTHIJKLMABCDEFG

then the words above would be sorted as

WHATEVER
ZONE
HOW
HOWEVER
HILL
ANY
ANTLER
COW

The first letters of the words are in their alphabetical order. Where words have the same prefix, the first differing letter determines the order, so the order goes ANY, then ANTLER,

since Y is before T in their choice of alphabet. Still HOWEVER comes after HOW, since HOW is a prefix of HOWEVER.

Dealing with the different alphabetical orders each year by hand (or tentacle) is tedious. Your job is to implement sorting with the English letters in a specified sequence.

Input:
The input will contain one or more datasets. Each dataset will start with a line containing an integer n and a string s, where s is a permutation of the English uppercase alphabet, used as the Gorellians' alphabet in the coming year. The next n lines (1 n 20) will each contain one non-empty string of letters. The length of each string will be no more than 30. Following the last dataset is a line containing only 0.

Output:
The first line of output of each dataset will contain "year " followed by the number of the dataset, starting from 1. The remaining n lines are the n input strings sorted assuming the alphabet has the order in s.

Sample Input	Sample Output
8 UVWXYZNOPQRSTHIJKLMABCDEFG	year 1
ANTLER	WHATEVER
ANY	ZONE
COW	HOW
HILL	HOWEVER
HOW	HILL
HOWEVER	ANY
WHATEVER	ANTLER
ZONE	COW
5 ZYXWVUTSRQPONMLKJIHGFEDCBA	year 2
GO	TEAMS
ALL	GO
ACM	GO
TEAMS	ALL
GO	ACM
10 ZOTFISENWABCDGHJKLMPQRUVXY	year 3
THREE	ZERO
ONE	ONE
NINE	TWO
FIVE	THREE
SEVEN	FOUR
ZERO	FIVE
TWO	SIX
FOUR	SEVEN
EIGHT	EIGHT
SIX	NINE
0	

E Carpet Squares

Data File: E.txt
Time allowed = 10 seconds

The Square Carpet Company makes hand-made carpet squares. They can make any size carpet as long as it is square and its side lengths are integer numbers of feet. These squares are so beautifully made that customers who want a rectangular shape will happily buy a number of squares and have them sewn together along their edges. For example, a room with width 12 and length 30 can be made from two 12x12 squares, plus two 6x6 squares.

Given the width and length of a rectangular shaped room as two integers both in [6,40], give the minimum number of carpet squares needed to cover the floor without gaps or overlaps.

Input:
The input file will begin with a single integer on a line by itself, T, giving the number of test cases to follow. Each test case will comprise a single line of text containing two space-separated integers, W and L, the width and length of the area to be carpeted.

Output: For each test case output one line containing an integer equal to the minimum number of carpet squares needed.

Sample Input	Sample Output
4	5
8 10	3
8 12	4
9 12	

F Travels in a square

Data File: F.txt
Time allowed = 10 seconds

An n x n game board is populated with integers, one nonnegative integer per square. The goal is to travel along any legitimate path from the upper left corner to the lower right corner of the board. The integer in any one square dictates how large a step away from that location must be. If the step size would advance travel off the game board, then a step in that particular direction is forbidden. All steps must be either to the right or toward the bottom. Note that a 0 is a dead end which prevents any further progress.

Consider the 4 x 4 board shown in Figure 1, where the solid circle identifies the start position and the dashed circle identifies the target. Figure 2 shows the three paths from the start to the target, with the irrelevant numbers in each removed.

2	3	3	1
1	2	1	3
1	2	3	1
3	1	1	0

Figure 1

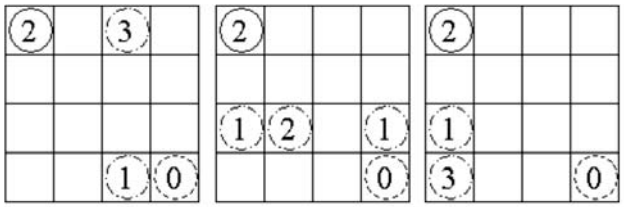

Figure 2

Input:
The input contains data for one to thirty boards, followed by a final line containing only the integer -1. The data for a board starts with a line containing a single positive integer n, $4 \leq n \leq 34$, which is the number of rows in this board. This is followed by n rows of data. Each row contains n single digits, 0-9, with no spaces between them.

Output:
The output consists of one line for each board, containing a single integer, which is the number of paths from the upper left corner to the lower right corner. There will be fewer than 263 paths for any board.

Sample Input	Sample Output
4 2331 1213 1231 3110 4 3332 1213 1232 2120 5 11101 01111 11111 11101 11101 -1	3 0 7