\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

BRONZE PROBLEMS

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Three problems numbered 11 through 13

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Problem 11: Scrabble [Traditional, 2009]

Bessie is playing Bovine Scrabble which is just like regular Scrabble

except that a player's tray has T (3 <= T <= 20) letters instead

of always having 7 letters as in the normal game.

For those who haven't played English Scrabble, it is a word game

where each player has a set of T letters and uses them to form a

word to place on a gameboard which ends up looking a little bit

like a crossword puzzle. For this task, Bessie has the first play

and cares only to make a word from one or more of her letters

(without worrying how to fit it onto the board).

Each letter has a value (see the table below). A word's value, for

this task's purposes, is the sum of values of each of the letters.

Thus, the word "TAX" has three letters, each with a value: "T"

with 1 point, "A" with 1 point, and "X" with 8 point -- total: 10

points. No other bonus points are considered in this task. Bessie

might use one, some, or all of her letters to make the highest

scoring word.

This game also includes blank letters (which are represented as "#"

in the input). Blank letters can be used to represent any other

letter but receives 0 points no matter which letter is chosen. Only

three of the test data inputs will contain one or more blanks. Each

blank letter can, if desired, represent a different actual letter

for purposes of forming words.

Given T and a set of T letters, read the dictionary from file

dict.txt (which is alphabetized and has fewer than 25,000 words;

each word is no longer than 20 characters) and determine which is

the most valuable word that can be made from Bessie's letters. If

two words have the same value, choose the one that comes first

alphabetically.

The actual dictionary used in this task is available at

http://ace.delos.com/usaco/dict.txt . Bessie can always find

some word in that dictionary to form with her letters.

Letter values:

0 points: # (blank)

1 point: A, E, I, L, N, O, R, S, T, U

2 points: D, G

3 points: B, C, M, P

4 points: F, H, V, W, Y

5 points: K

8 points: J, X

10 points: Q, Z

PROBLEM NAME: scrab

INPUT FORMAT:

\* Line 1: A single integer: T

\* Lines 2..T+1: Each line has a single character that appears in

Bessie's letter tray

SAMPLE INPUT (file scrab.in):

3

A

T

X

OUTPUT FORMAT:

\* Line 1: A single line with a single word (the word -- no '#' blanks)

that is the highest-scoring word Bessie can make.

SAMPLE OUTPUT (file scrab.out):

TAX

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Problem 12: Back to the Barn [Neal Wu, 2008]

As every cow does occasionally, Bessie has lost herself in the

woods! She desperately needs to get back to the barn but has no

idea where to go.

The woods can be thought of as an R x C grid (1 <= R <= 5; 1 <= C

<= 5). Bessie is located in the lower left corner at row 1, column

1; the barn is located in the upper right corner at in row R, column

C. Each square in the grid is either empty (denoted by '.') or

blocked by a tree (denoted by 'T'). Bessie's square and the barn

will always be empty.

From a given square, Bessie may move to any adjacent square (one

that shares a long edge with Bessie's current square) as long as

it is empty. As Bessie is quite a smart cow, she never visits a

square twice on the way back to the barn.

Determine the number of different ways Bessie can take that lead

her from her initial position back to the barn while visiting at

most K different squares (1 <= K <= R \* C).

By way of example, consider this forest:

....

.T..

....

Bessie has a number of ways to get to the upper right corner, here

are all seven of them and their path lengths (the number of squares

visited):

cdef ...f ..ef ..gh cdeh cdej ...f

bT.. .T.e .Td. .Tfe bTfg bTfi .Tde

a... abcd abc. abcd a... a.gh abc.

Length: 6 6 6 8 8 10 6

PROBLEM NAME: backbarn

INPUT FORMAT:

\* Line 1: Three space-separated integers: R, C, and K

\* Lines 2..R+1: Line i+1 contains the C characters representing row

R+1-i of the forest (with no spaces)

SAMPLE INPUT (file backbarn.in):

3 4 6

....

.T..

....

INPUT DETAILS:

The forest is a 3 x 4 grid. A single tree sits almost in the middle,

as shown. No more than six steps can be taken.

OUTPUT FORMAT:

\* Line 1: A single integer that is the number of different paths,

visiting at most K squares, that Bessie can take to get back

to the barn. Note that this number will always fit into a

signed 32-bit integer.

SAMPLE OUTPUT (file backbarn.out):

4

OUTPUT DETAILS:

Of the several evenays to complete the traversal, only four of them touch

no more than six squares along the route.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Problem 13: Music Notes [Neal Wu, 2008]

FJ is going to teach his cows how to play a song. The song consists

of N (1 <= N <= 100) notes, and the i-th note lasts for B\_i (1 <=

B\_i <= 100) beats. The cows will begin playing the song at time 0;

thus, they will play note 1 from time 0 to time B\_1 - 1, note 2

fromtime B\_1 to time B\_1 + B\_2 - 1, etc.

The cows have lost interest in the song, as they feel that it is

long and boring. Thus, to make sure his cows are paying attention,

FJ asks them Q (1 <= Q <= 1,000) questions of the form, "During the

beat at time T\_i (0 <= T\_i < length of song), which note should you be

playing?" The cows need your help to answer these questions.

By way of example, consider a song with these specifications: note

1 for length 2, note 2 for length 1, and note 3 for length 3:

NOTES 1 1 2 3 3 3

+---+---+---+---+---+---+

TIME 0 1 2 3 4 5

PROBLEM NAME: mnoteb

INPUT FORMAT:

\* Line 1: Two space-separated integers: N and Q

\* Lines 2..N+1: Line i+1 contains a single integer: B\_i

\* Lines N+2..N+Q+1: Line N+i+1 contains a single integer: T\_i

SAMPLE INPUT (file mnoteb.in):

3 5

2

1

3

2

3

4

0

1

INPUT DETAILS:

The song has 3 notes, with lengths 2, 1, and 3. FJ has 5 queries.

OUTPUT FORMAT:

\* Lines 1..Q: Line i contains the a single integer that is the note

the cows should be playing at time T\_i

SAMPLE OUTPUT (file mnoteb.out):

2

3

3

1

1

OUTPUT DETAILS:

As per the chart above.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*