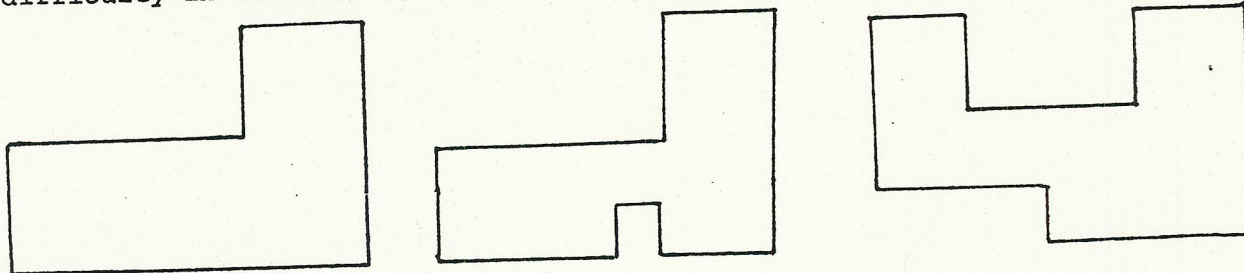


TEXAS REGIONAL PROGRAMMING CONTEST

April 6, 1974

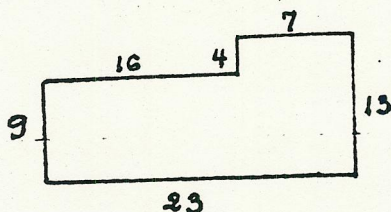
Upper Division - Problem 1

HARDFLOR, Inc. is in the business of laying terrazzo floors. They charge by the square foot of flooring laid. They are very good at laying floors, but they have some difficulty in calculating areas because floors come in many shapes and sizes, e.g.,



One rule that John Hardflor, the company's founder, made years ago was to "never bid on a job that doesn't have square corners." And, the company still follows that rule. They are still having trouble estimating areas, however, and they would now like to have a computer program developed which would assist them in this endeavor.

The company's estimator would like to go out on a potential job and measure the dimensions of the room and code them in the following way: He starts at an arbitrary corner. He then lists a succession of direction distance pairs. (Directions are always N, S, E, W and distances are always in feet). e.g. A room with dimensions as shown



might be coded with the following set of pairs: [(N,9) (E,16) (N,4) (E,7) (S,13) (W,23)]

Write a program which will determine the area. Assume that the coding is always correct--that you always end up at the corner on which you started. Assume you never have more than 12 corners in a room. A room description will be contained on a single card with the format: (1X, I2, 2X, I2 (A1, I2, 2X)).

The "number of corners" will be contained in columns 2 and 3 of the card and successive direction-distance pairs will be contained in columns 5-7, 10-12, 15-17, etc. There may be any number of room descriptions, but there will be a blank card after the last one. For each description, print out the area exactly as shown below:

THE AREA IS 99999



TEXAS REGIONAL PROGRAMMING CONTEST

April 6, 1974

Upper Division - Problem 2

This problem requires that you write a program which does three tasks which might facilitate the use of a communication line.

The first task is that of translating from one set of codes to another. The second task is to compress 80-character card images to a smaller size by compressing blanks. The final task is that of grouping these "compressed" records into larger records. The three tasks are to be performed in sequence.

Your program will have one control card which will be followed by an arbitrary number of data cards. The last card will have asterisks in columns 1 - 4. The control card will have the following format:

<u>Columns</u>	<u>Contents</u>
1 - 4	N, the length of the "new" records, $N < 1000$
6 - 7	M, the number of characters to be translated, $M \leq 20$
10 - (9 + M)	The M distinct characters to be translated
40 - (39 + M)	The M characters to which the first M characters are to be translated.

For example, a control card containing:

0400 03 ABC \$,\$

means that the program is to produce 400 character records and that all the A's and C's are to be converted to "\$" and B's are to be converted to ",".

Characters which appear on the data cards but not on the translate list will be unaffected.

Compression is effected by replacing each string of 4 or more blanks by an ampersand, "&", followed by two digits containing the number of blanks. Thus, the compressed form of a translated card containing ABC in columns 10 - 12 and nothing else would be :           &09ABC&68

Finally, after translation and compression have taken place the records should be grouped into "new" records of length N by concatenating compressed records until the "new" record is full. No compressed record should be split between two "new" records whenever it is impossible to add an additional compressed record to a "new" record, then the remainder of the "new" record should be filled with blanks.

Advance to a new page and write out the "new" records on the printer, doublespaced, using as many lines as necessary to represent each record in columns 10 - 109 of each line. No extra blank lines between records are needed.

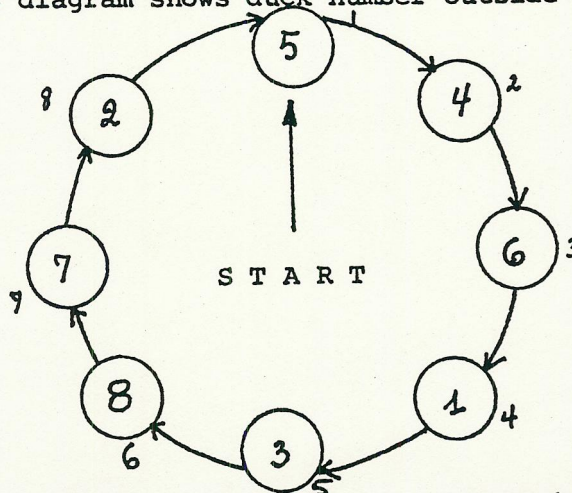


TEXAS REGIONAL PROGRAMMING CONTEST

April 6, 1974

Lower Division - Problem 3

Suppose there are  $n$  ducks floating on the pond in a circle. The pond is also home for an alligator with a fondness for ducks. Beginning at a particular position (duck number 1) the alligator counts around the circle and eats every  $m$ th duck (the circle closing as ducks are eaten). For example, when  $n = 8$ , and  $m = 4$ , the following diagram shows duck number outside each node and consumption order inside:



The first duck is fifth on the menu, the second is fourth, etc. The sequence 5 4 6 1 3 8 7 2 of orders of consumption completely describes the alligator's menu.

Write a program which prints out the orders of consumption of the ducks given  $n$  and  $m$  on an input card in the following format:

```
c 2 - 3      n
c 5 - 6      m
```

There may be multiple input cards. Neither  $n$  nor  $m$  will be zero. A blank card marks the end of the data. The output should begin on a new page at the left side line, use 3 columns for each integer, and continue on subsequent lines as needed. Double space between the output produced by the individual cases and begin again at the left of the printed line each time.

1980-91 ACM SOUTHEAST REGIONAL PROGRAMMING CONTEST

FREE-FORM INPUT

You are to read a set of data cards and, for each data card, compute the sum of the numbers on that card. Each card contains 1 or more numbers in columns 1-72. Numbers are separated by commas. A number may be positive or negative (preceded by a minus sign), and may or may not contain a decimal part. Blanks may be imbedded between numbers and within numbers. An example



1 23.4,-5 , .00 2 , 1 8 6

The numbers on this card are 123.4, -5, 0.002, and 186.

The output for each card should consist of the card itself and the resulting sum, e.g.,

```
CARD=1 23.4,-5 , .00 2, 1 8 6
SUM=          304.402000000
```

You may use a F20.9 format for the sum. The program should halt when a card containing all blanks in columns 1-72 is encountered.

NOTES:

- 1) You may assume correct input.
- 2) You may assume that all values can be represented as standard real numbers.
- 3) You need not worry about the precision of fractional numbers; 0.1 cannot be represented exactly but the internal representation will be accepted as correct.

1980-81 ACM East Central Regional Programming Contest  
TRIANGULAR TEXT

Write a program which will read a text file and print the text in a triangle. The rows of the triangle are to be single spaced and left justified. The last row of the triangle is to contain one word. Periods are to be included with the final word of each sentence and are the only type of punctuation in the input text. Each row of the triangle is to contain the minimum number of words needed in order to make that row longer than the row below. Any excess words are to be placed in the first row. The text contains at most 400 characters. Each input record is in FORMAT(80A1), no extra blanks appear between words or before the first word, and no word is split across cards. The end of input is indicated by a blank record. An example of input and output follows:

input

MUCH CURRENT RESEARCH IN ARTIFICIAL INTELLIGENCE INVOLVES  
DESIGNING PROGRAMS THAT CAPTURE THE KNOWLEDGE AND REASONING  
PROCESSES OF HIGHLY INTELLIGENT SPECIALISTS. THE PRACTICAL GOAL OF  
SUCH WORK IS TO MAKE SPECIALIZED EXPERTISE MORE GENERALLY  
ACCESSIBLE.

output

MUCH CURRENT RESEARCH IN ARTIFICIAL INTELLIGENCE INVOLVES  
DESIGNING PROGRAMS THAT CAPTURE THE KNOWLEDGE AND  
REASONING PROCESSES OF HIGHLY INTELLIGENT  
SPECIALISTS. THE PRACTICAL GOAL  
OF SUCH WORK IS TO MAKE  
SPECIALIZED EXPERTISE  
MORE GENERALLY  
ACCESSIBLE.