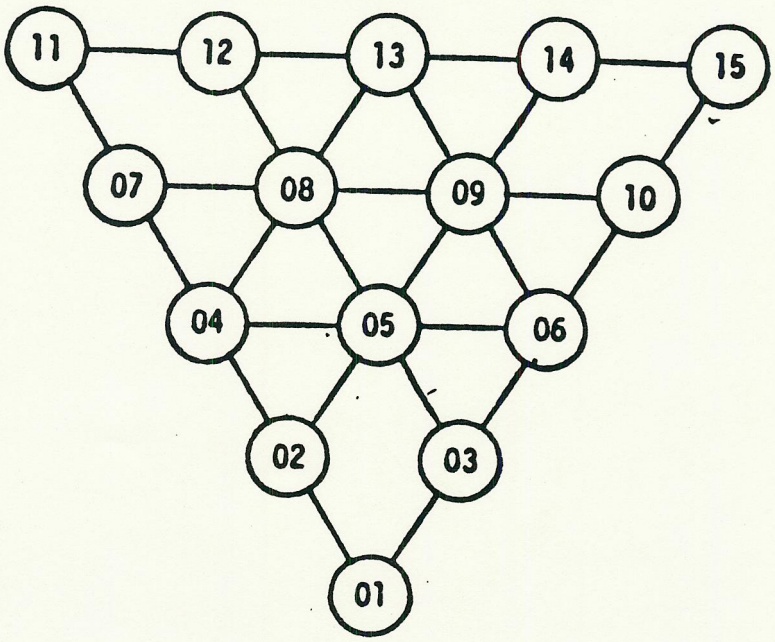
1979 NATIONAL SCHOLASTIC PROGRAMMING CONTEST

PROBLEM A

PEG PUZZLES

A little puzzle which has become popular at various restaurants consists of a triangular piece of wood with 15 holes drilled into it. The numbering and arrangement of these holes is as follows:



The puzzle begins with wooden pegs in 14 of the holes. The goal is to "jump and remove" 13 of these 14 pegs, ending with only one peg remaining on the board. The rules are as follows:

1) a peg can be moved only by jumping, and hence removing, a single adjacent peg;
2) "adjacent" means "connected by one of the lines drawn on the board";
3) the jump must be made entirely along the connecting line and its extension;
4) the jump can be made only if the hole on the other side of the peg to be jumped is vacant.

This gives 36 potential moves which are tabulated on the following page.

peg at	jumps peg at	and moves to vacant hole
01	02	04
04	02	01
01	03	06
06	03	01
02	04	07
07	04	02
03	06	10
10	06	03
04	07	11
11	07	04
06	10	15
15	10	06
11	12	13
13	12	11
12	13	14
14	13	12
13	14	15
15	14	13
02	05	09
09	05	02
03	05	08
08	05	03
04	05	06
06	05	04
04	08	13
13	08	04
05	08	12
12	08	05
07	08	09
09	08	07
05	09	14
14	09	05
06	09	13
13	09	06
08	09	10
10	09	08

The program is to read an initial board configuration. This configuration is described by 15 binary digits, format (15I1); a "1" indicates that the hole corresponding to that column has a peg in it, a "0" (zero) indicates that it is empty. Since the puzzle starts with 14 pegs and one empty hole, there will be 14 columns with "1"s in them and one with a "0". The program is then to compute and print a sequence of moves which will terminate with one peg remaining on the board. Such a sequence will exist for the initial configurations the program will encounter in the "judged run" data deck. Format for this output is (7H PEG AT,I3,13H JUMPS PEG AT,I3,13H AND MOVES TO,I3).

SAMPLE INPUT:

```
111101111111111
```

SAMPLE OUTPUT:

```
PEG AT 14 JUMPS PEG AT  9 AND MOVES TO  5
PEG AT  7 JUMPS PEG AT  8 AND MOVES TO  9
PEG AT 10 JUMPS PEG AT  9 AND MOVES TO  8
PEG AT 12 JUMPS PEG AT 13 AND MOVES TO 14
PEG AT  3 JUMPS PEG AT  6 AND MOVES TO 10
PEG AT 15 JUMPS PEG AT 10 AND MOVES TO  6
PEG AT  2 JUMPS PEG AT  4 AND MOVES TO  7
PEG AT  6 JUMPS PEG AT  5 AND MOVES TO  4
PEG AT  7 JUMPS PEG AT  4 AND MOVES TO  2
PEG AT  1 JUMPS PEG AT  2 AND MOVES TO  4
PEG AT  4 JUMPS PEG AT  8 AND MOVES TO 13
PEG AT 14 JUMPS PEG AT 13 AND MOVES TO 12
PEG AT 11 JUMPS PEG AT 12 AND MOVES TO 13
```

The programs is then to skip 3 or more blank lines and read another initial configuration. If it is all zeroes, stop. Otherwise, solve its game and continue.

1979 NATIONAL SCHOLASTIC PROGRAMMING CONTEST
PROBLEM B
REPEATING DECIMALS

Where x and y are integers, the rational number x/y has a decimal equivalent of the form

iiii.fffff...

If the f's are zero from some point onward, they are normally omitted and an exact decimal equivalent can be written. Otherwise, the sequence of fraction digits (f's) is infinite but will ultimately become repetitions of some subsequence. Denoting the repeating subsequence with r's, the repeating decimal representation uses the notation

$$iii.fffff\overline{r_1 \ldots r_k}$$

to convey the infinite decimal equivalent

$$iii.fffff r_1 \ldots r_k r_1 \ldots r_k r_1 \ldots r_k \ldots$$

For example,

$$\frac{22}{7} = 3.\overline{142857} \qquad \frac{5}{12} = 0.41\overline{6} \qquad \frac{100}{11} = 9.\overline{09}$$

Notice that the repeating group is the leftmost such subsequence to the right of the decimal point, in the fraction digits only.

Write a program which will:

1) Read a card containing two positive integers in format 2I10, say x and y.

2) Skip two lines and print a three-line group indicating the decimal equivalent of x/y in repeating decimal format. If more than 100 digits to the right of the decimal point would be required, print a line showing the values x and y and a message indicating the difficulty. If the rational number has an exact finite decimal equivalent, print it without a repeating group or trailing zeroes to the right of the decimal point. Examples are given below in the required format.

3) Repeat from step 1) until a card having y=0 is encountered.

$$\frac{27836}{462} = 60.251082 \qquad \frac{1}{252} = 0.00396825$$

$$\frac{2}{17} = 0.1176470588235294 \qquad \frac{300030003}{16} = 18751875.1875$$

↑ up to nine extra blanks are permitted here

1979 NATIONAL SCHOLASTIC PROGRAMMING CONTEST

PROBLEM C

EMBEDDED ARABIC TO ROMAN

The Roman emperor has just received a message from an English king. However, the emperor does not understand Arabic numerals. Write a program to translate all numeric values into Roman numerals, leaving the character strings preceding and following the numerals unchanged. All numeric values will be less than 4000_{10} and contain no embedded punctuation. The message may be many cards in length and words may be split across card boundaries, although numerals will not be split by card boundaries. All 80 columns of each card may be used. Luckily, the number 0 does not appear. The message is terminated by the pair ##. Print one line corresponding to each card, with the Arabic numerals transformed to Roman and surrounded text unchanged.

e.g.

PLEASE ADVISE BY 19 OCTOBER, 1001.

is to be changed to read

PLEASE ADVISE BY XIX OCTOBER, MI.

Arabic	Roman
1	I
2	II
3	III
4	IV
5	V
6	VI
7	VII
8	VIII
9	IX
10	X
20	XX
30	XXX
50	L
90	XC
100	C
140	CXL
500	D
1000	M

1979 NATIONAL SCHOLASTIC PROGRAMMING CONTEST

PROBLEM D

SPEEDOMETER CORRECTIONS

An automobile speedometer is connected to the transmission of a car and transforms the number of revolutions per second of the drive shaft to a speed in miles per hour. But, the size of the tire changes from time to time, it gets smaller as the tread wears down, it gets larger when it is hot, its size depends on the amount of air in the tire, and so forth. Consequently, even if the speedometer accurately measures the number of revolutions per second, its reading might be different from the speed that the car is traveling. Turnpikes and some of the interstate highways have mileposts along the roadway. If a driver keeps the car traveling at a fixed speedometer reading and if a passenger times the intervals between mileposts, then one can determine the actual speed of the car. The program is to print the seconds between mileposts for a car traveling at speeds of 10, 20, 30, 40, 50, 60, 70 and 80 miles an hour, rounded to the nearest second. (The distance, d, is related to the velocity, v, and the time, t, according to d=v*t.) The velocity is in miles per hour, so you will have to multiply the velocity in miles per hour by the appropriate factor to change it to miles per second.

The rolling diameter of a tire is about 29 inches for a full-sized station wagon and about 25 inches for a small compact car. The circumference is equal to diameter times π. The program is to determine and print the number of revolutions each tire makes when the car goes one mile (there are 5280 feet in a mile), rounded to the nearest revolution. The program should also determine and print the number of revolutions per second of each of the two tires when the car is going, 30, 50, 70 and 90 miles per hour. π = 3.14159265358979323 85.

Suppose that the diameter of the tire is -10 percent, -5 percent, -1 percent, +1 percent, +5 percent, or +10 percent different from the diameter assumed in the speedometer calibration. The program is to determine and print the actual speed when the speedometer reads 50, 60, 70, 80 and 90 miles per hour, for each of these errors and each tire size to the nearest tenth of a mile per hour.

All output is to be clearly labeled. Use a tabular format whenever possible.