ACM ICPC 97
Asia Regional Programming Contest
The Twenty-second Annual ACM International Collegiate Programming Contes
(Asia Region/Taipei)

Problem A

Jugs Puzzle

Input file: Jugs.in

Problem Description

There are two jugs, A and B, and an infinite supply of water. There are three types of actions that you can take:

(1) Fill a jug
(2) Empty a jug, and
(3) Pour from one jug to the other.

The action of pouring from one jug to the other stops when the first jug is empty or the second jug is full, whichever comes first. ***For example, both jug A and B has capacity of 8 gallons. If jug A has 5 gallons of water and B has 6 gallons of water, then pouring from A to B will fill up jug B and leave 3 gallons of water in jug A.*** Please develop a program to solve a puzzle given by a triple (Ca, Cb, N), where Ca and Cb are the capacities of the jugs A and B, respectively, and N is the goal, representing the mount of water left in jug B. The possible steps are:

- Fill A
- Fill B
- Empty A
- Empty B
- Pour A B
- Pour B A
- Success

Where "Pour A B" means "pour the contents of jug A into jug B", and "Success" means that the goal has been accomplished.

Example

In the movie "Die Hard 3", Bruce Willis and Samuel L. Jackson were give the following puzzle to solve. They were given a 3-gallon jug and a 5-gallon jug and were asked to fill up the 5-gallon jug, and exact 4 gallons of water in 5-gallon jug. This problem can be expressed in a triple as (3,5,4), and solved in the following way:

Jug A
(capacity=3 gallons) Jug B
(capacity= 5 gallons) Goal is to get exactly 4 gallons in jug B.
Fill A
Pour A B
Fill A
Pour A B
Empty B
Pour A B
Fill A

Pour A B
Success

Notes

1. You may assume that the input you are given does have a solution.
2. The output of your program is a sequence of steps that leaves exactly N gallons in jug B.
3. Both jug A and jug B are initially empty.

Input Format

(1) The input file name is “jugs.in.” It consists of a series of input lines each defining one puzzle.
(2) Input for each puzzle is a single line of three positive integers: Ca, Cb, and N. Ca and Cb are the capacities of jugs A and B, and N is the goal.
(3) You can assume $0 < Ca \leq Cb$ and $N \leq Cb \leq 1000$.

Output Format

(1) Output from your program should be printed onto the screen. The output consists of a series of instructions from the list of the potential output lines which will result in either of the jugs containing exactly N gallons of water.
(2) The last line of output for each puzzle should be the line “Success”. Output lines start in column 1 and there should be no empty lines nor any trailing spaces.

Sample Input

If there are two puzzles in the input-file (jugs.in). The content of the file should look like the two lines below:

```
5 7 3
11 13 9
```

Sample Output

```
Fill A
Pour A B
Fill A
Pour A B
Empty B
Pour A B
Success
Fill A
Pour A B
Fill A
Pour A B
Empty B
Pour A B
Success
```

ACM ICPC 97
Asia Regional Programming Contest
The Twenty-second Annual ACM International Collegiate Programming Contes
(Asia Region/Taipei)

Problem B

Filling Water Pot

Input file: pot_in.txt

Problem Description

Suppose that you have a water pot with a capacity of L liters and n water bottles with capacities c_1, c_2, ..., c_n. Write a program to select *m* bottles of water to fill up the water pot, or answer there is no solution. The selected bottles have to be arranged in increasing order. If you select a bottle of water, the bottle has to be completely poured in. Also, all bottles are full initially. If two solutions have the same numbers of bottles, select the one that has a bottle with the largest capacity. For example, if {5, 8} and {6, 7} are two possible solutions to fill up a water pot of capacity 13, then select {5, 8} as the problem solution. If the largest volume in two solutions is the same and there is more than one possible solution, you have to select the one with fewer bottles. For example, if {2, 3, 8} and {5, 8} are two possible solutions to fill up a water pot with capacity 13, then you have to select {5, 8} as the problem solution. Note that if you try to solve this problem by a brute-force method, the program execution time may exceed the run-time limit.

Input Format

A sequence of pot and bottle volumes will be given in file "pot_in.txt". For each set, the first line gives the pot capacity L. Then a sequence of numbers representing the capacity of bottles will be given in each of the following lines except the last line. The end of sequence is signified by `-1`. Note that all capacities given are integer numbers.

Output Format

Output the answer of each set of pot and bottles in the input file to file "pot_out". If the *n*th set has a solution, then print "`Solution of set` *n* `is`" and then the bottle capacities in an increasing order one by one. If there is no answer, then print out "`Set` *n* `has no solution.`" in a single line.

Sample Input

```
350
421 3 13 92 41 63 84 14 29 37
20 102 50 10 ®C1
13
3 2 8 5 4 -1
```

Sample Output

```
Solution of set 1 is: 14 29 50 63 92 104
Solution of set 2 is: 5 8
```

ACM ICPC 97
Asia Regional Programming Contest
The Twenty-second Annual ACM International Collegiate Programming Contes
(Asia Region/Taipei)

Problem C

Component Labeling

Input file: picture.txt

Problem Description

In the applications of image processing, we may convert the input gray-level pictures (captured from CCD camera) are usually converted into binary pictures for object identification. In a binary picture, we often treat each individual connected component as a separate object. A connected-component is defined as follows. Given a point P (located at (i, j)), its upper (located at (i, j-1)), lower (located at (i, j+1)), left (located at (i-1, j)), and right (located at (i+1, j)) neighbors are its 4-connected neighbors. The 4-connected neighbors of the point P are illustrated in the following 3×3 sub-picture:

(i, j-1)
(i-1,j) (i, j) (i+1,j)
(i, j+1)

A component C consists of all the points with value 1, in which each point has at least one 4-connected neighbor in C. In the following figure, the left part represents an input picture, and the right part represents an output picture. Two components are found and labeled by characters a and b, respectively.

```
0000000000  0000000000
0011111100  00aaaaaa00
0010000100  00a0000a00
0011111100  00aaaaaa00
0000000011  00000000bb
```

Write a program to differentiate different components in a binary picture by labeling each component uniquely. You can label a component by any alphabet.

Input Format

The input file is “picture.txt”, which may contain several pictures. A picture is represented by a 2-D integer array and then 999 in the last line.

Output Format

Print your results to file “label.txt”. Different components have to be labeled by different characters. The end of a labeled picture is represented as 999 in a separate line.

Sample Input

```
00001110011011
00111011110001
11001111011110
999
00111111111100
00100000000100
00101111110100
00101000010100
00101011010100
00101011010100
00101000010100
00101111110100
00100000000100
00111111111100
999
```

Sample Output

```
0000aaa00aa0cc
00aaa0aaaa000c
ee00aaaa0aaaa0
999
00aaaaaaaaaa00
00a00000000a00
00a0bbbbbb0a00
00a0b0000b0a00
00a0b0cc0b0a00
00a0b0cc0b0a00
00a0b0000b0a00
00a0bbbbbb0a00
00a00000000a00
00aaaaaaaaaa00
999
```

ACM ICPC 97
Asia Regional Programming Contest
The Twenty-second Annual ACM International Collegiate Programming Contes
(Asia Region/Taipei)

Problem D

Reordering Cars on a T-shape Railway

Input file: traininp.dat

Problem Description

A T-shape railway with three distinct locations represented by three points X, Y and Z respectively (as shown in the figure), was designed to rearrange the order of cars in a train. A train is initially stay at the starting point X. The order of its cars are labeled in alphabetically order A, B, C, ... from left to the right. The cars are to be reordered according to a pattern specified in the input file. For example five cars of a train ABCDE many be reordered as BADCE, where BACDE is specified in the input file. In order to reorder the cars, each of them may move from point X to point Z. Or, the car can move from point X topoint Y, then wait for some other cars behind it to move directly from X to Z before it proceeds to Z. No more than three cars can stay in point Y at the same time. None of the following situations are allowed: Z toX, Z to Y, or Y to X. Design a program that lists the steps of reordering the cars of a train.

Input Format

Each line of the input file "traininp.dat" contains a string which shows the pattern for a car rearrangement case. According to the pattern, the cars are reordered when they arrived at point Z.

Output Format

Print the result on the screen and save it as an output file "trainopt.dat". The file should include three columns: the first column shows the moving step, the second column indicates the alphabet of the car to be moved, and the third column shows the points where the car goes. Leave a space line between each reordering case. Print "`No Solution`" in case that there is no way to reach the given pattern.

Sample Input

```
BADCE
CBA
DBAC
ABCD
```

Sample Output

```
1 A X->Y
2 B X->Z
3 A Y->Z
4 C X->Y
5 D X->Z
6 C Y->Z
7 E X->Z

1 A X->Y
2 B X->Y
3 C X->Z
4 B Y->Z
5 A Y->Z

No Solution

1 A X->Z
2 B X->Z
3 C X->Z
4 D X->Z
```

ACM ICPC 97
Asia Regional Programming Contest
The Twenty-second Annual ACM International Collegiate Programming Contes
(Asia Region/Taipei)

Problem E

Negative-base Numbers

Input file: q5.in

Problem Description

When we write decimal (base 10) numbers, we use a positional notation; each digit is multiplied by an appropriate power of 10 depending on its position in the number. Similarly, for binary (2) base numbers, each binary digit is multiplied by an appropriate power of two. In general, any positive integer R or negative integer -R can be chosen as the base of a number system. If the base is R or -R then R digits (0, 1, . . . R-1) are used. For example, if R=7 then the required digits are 0, 1, 2, 3, 4, 5 and 6, no matter they are for R or -R. For bases greater than 10 or less than -10, more than 10 symbols are needed to represent the digits. In this case, English characters are usually used to represent digits greater than 9. For example, in hexadecimal (base 16) numbers, A represents 10 (base 10), B represents 11 (base 10), C represents 12 (base 10), D represents 13 (base 10), E represents 14 (base 10) and F represents 15 (base 10). A negative-base number uses -R as the base. For example -15 (base 10) is equivalent to 1 (base -2) and its power series in -2 can be expressed as $1*(-2)^5+1*(-2)^4+0*(-2)^3+0*(-2)^2+0*(-2)^1+1*(-2)^0$.

Please design and implement a program which reads in a decimal integer number (base 10) and a negative base $-R \in 12\{-2, -3, -4, \ldots -9, -10, -11, -12, -13, -14, -15, -16\}$, and then converts the decimal integer into a negative-base number.

Input Format

The input file name is “q5.in”. It contains several lines. There are two input data in each line. The first one is a decimal integer n where $-32{,}768 \le n \le 32{,}767$; the second one is a negative number -R which is used as the negative base, where $-R \in 12\{-2, -3, -4, \ldots -9, -10, -11, -12, -13, -14, -15, -16\}$. A line containing only the character, ‘#’, marks the end of the input file.

Output Format

The output data should be printed onto the screen. For each input data, print the negative-base numbers and the negative bases. If the base is less than -10, more than 10 symbols are needed to represent the digits. For example, in base -16, A represents 10 (base 10), B represents 1 (base 10), C represents 12 (base 10), D represents 13 (base 10), E represents 14 (base 10) and F represents 15 (base 10). The output data of the program should be shown on the screen.

Sample Input File

```
30000 -2
-20000 -2
28800 -16
-25000 -16
#
```

Sample Output File

```
30000=11011010101110000 (base -2)
-20000= 1111011000100000 (base ®C2)
28800= 19180 (base ®C16)
-25000= 7FB8 (base -16)
```

ACM ICPC 97
Asia Regional Programming Contest
The Twenty-second Annual ACM International Collegiate Programming Contes
(Asia Region/Taipei)

Problem F

Verify a Conjecture

Input file: line.d

ACM ICPC 97
Asia Regional Programming Contest
The Twenty-second Annual ACM International Collegiate Programming Contes
(Asia Region/Taipei)

Problem G

Basketball Team Assignment

Input file: team.txt

Problem Description

There are n students with a special relation among them in a university. Based on this special relation, they are trying to organize several basketball teams. This special relation can be specified as follows:

The students are numbered from 1 to n. If student i has a special relation with student j, student j also has the special relation with student i. Hence, the existence of the special relation between students i and j can be denoted as a pair (i, j).

Each team possesses the following properties:

1. Each team has exactly 5 students. Any two students in the same team must have the special relation between them.
2. Each team is the maximal group in which any two students must have the special relation. That is, it is impossible to have a student not in the team and he/she has the special relation to every student in the team.

Example: (a) Let n=6 and the special relation be specified by a graph which is given in Fig. (a). If two students have the special relation, then there is an edge between them. We can find that {1, 2, 3, 4, 5} forms a team.

(Lost picture)
Fig. (a)

(b) Another special relation is specified in Fig. (b). We can not find a team with 5 students. For instance, if we select {1, 2, 3, 4, 5}, then student 6 is not in the team, but he/she has the special relation with every student in the team.

(Lost picture)
Fig. (b)

Now, the university decides to select a 5-member team from those teams with the highest average height to represent the university to join a basketball tournament. If two teams have the same average height, then the 5-member team, which contains the student with the smallest number will be selected. For example, if {2, 3, 5, 6, 9} and {4, 5, 9, 10, 11} are both 5-member teams with the same highest average, then {2, 3, 5, 6, 9} will be selected due to the student with the smallest ID ‘2’ in that team.

Input

The input file name is “team.txt”. There are two parts of the input file. The first part of the input file contains all the pairs specified the special relation among those n students, for instant (i, j) which indicates students i and j having the special relation. A (0, 0) is used to indicate the end of the first part. Note that all the students from 1 to n will appear at least once in the first part, hence the value of n can be obtained by scanning the input data in this part. The second part of the input file contains the height of those n students from student 1 to student n.

Output

The output of your program should be printed onto the screen, and it may be redirected into a printer. Print exactly the information stated below, with no cursor controls. If there are no 5-member teams, then output “NO”, otherwise, output the numbers of the 5 selected students and also the average height of the team.

Sample Input

```
(1, 2), (1, 3), (1, 4), (1, 5), (2, 3), (2, 4), (2, 5),
(3, 4), (3, 5), (4, 5), (2, 6), (3, 6), (4, 6),
(5, 6), (1, 7), (2, 7), (6, 7), (0, 0),
166, 167, 190, 199, 188, 166, 170
```

Output for the Sample Input

```
The team selected is {1, 2, 3, 4, 5}
The average height is equal to 182.
```

ACM ICPC 97
Asia Regional Programming Contest
The Twenty-second Annual ACM International Collegiate Programming Contes
(Asia Region/Taipei)

Problem H

Sum of Consecutive Positive Integers

Input file: q8input.dat

Problem Description

Design a program to input a positive integer x, where x>0. For the integer x, the program has to convert it into the sum of consecutive positive integers. For example, the sum of consecutive positive integers for integer 10 is "10=1+2+3+4". In addition, the total number of consecutive positive integers in the sum expression should be maximal. For example, the integer 9 can be expressed as "9=2+3+4" or "9=4+5". But only the sum expression "9=2+3+4" is valid. Some integers, such as integer 4, cannot be represented as a sum of consecutive positive integers. In this case, the output of your program should be "`4: (NO ANSWER)`".

Input Format

The program has to receive a series of integers in the file "q8input.dat". A comma separates two integers. Each integer has to be converted into a sum expression with maximal consecutive positive integers.

Output Format

1. The output should be able to display on the screen as well as print out to the printer. In the output, two expressions should be displayed in different lines.
2. If the integer x cannot be represented as a sum of consecutive positive integers, the expression "`x: (NO ANSWER)`" is output.

Sample Input

```
9,38,250,255, 256,260,468,528,876,1000
```

Sample Output

```
9=2+3+4
38=8+9+10+11
250=3+4+5+6+7+8+9+10+11+12+13+14+15+16+17+18+19+20+21+22
255=7+8+9+10+11+12+13+14+15+16+17+18+19+20+21+22+23
256: (NO ANSWER)
260=14+15+16+17+18+19+20+21+22+23+24+25+26
468=8+9+10+11+12+13+14+15+16+17+18+19+20+21+22+23+24+25+26+27+28+29+30+31
528=1+2+3+4+5+6+7+8+9+10+11+12+13+14+15+16+17+18+19+20+21+22+23+24+25+26+27+28+29+30+31+32
876=25+26+27+28+29+30+31+32+33+34+35+36+37+38+39+40+41+42+43+44+45+46+47+48
1000=28+29+30+31+32+33+34+35+36+37+38+39+40+41+42+43+44+45+46+47+48+49+50+51+52
```