Tasks

Task	A (product)	B (stones)	C (strings)	D (queen)	E (mary)	F (bridge)	G (cnm)	H (digit)
Sample test	A (product)	B (stones)	C (strings)	D (queen)	E (mary)	F (bridge)	G (cnm)	H (digit)
Time Limit(sec)	1	1	1	1	1	1	1	1

Problem A. Product

There is an array of **N** integer numbers in the interval from –30000 to 30000. The task is to select **K** elements of this array with maximal possible product.

Input data: the text file PRODUCT.IN consists of **N+1** lines. The first line contains **N** and **K (**$1 \le K \le N \le 100$) separated by one or several spaces. The others contain values of array elements.

Output data are located in a text file PRODUCT.OUT containing a single line with values of selected elements separated by one space. These values must be in non-increasing order.

An example of the input data

4 2

1

7

2

0

An example of the output data

7 2

Problem B. Dropping the stones

You have **N** stones (1 £ **N** £ 10) , each stone is characterized by weight $\mathbf{p_i}$ and cost $\mathbf{v_i}$ **(i=1, ..., N)** . You should drop these stones by a gutter into one of the two bunkers: A or B.

The mechanism of receiving bunkers switching works as follows. The stones are dropping into one of the bunkers until the weight of this bunker's content will exceed the weight of stones in other bunker by at least **D**. Then the gutter switches to the other bunker. At the beginning both bunkers are empty, and the first stone falls into the bunker A.

The task is to collect stones with maximum total cost in bunker B after all stones have been dropped.

Input data : the text file STONES.IN consists of **N+1** lines. The first line contains values **N** and **D** separated by one or several spaces. The next lines contain values $\mathbf{p_i}$ and $\mathbf{v_i}$, also separated by spaces. All input data (except **N**) are integers from 0 to 10000.

Output data are located in a text file STONES.OUT containing a single line with total cost of stones in the bunker B.

An example of the input data

4 2

2 2

2 2

1 1

1 1

An example of the output data

3

Problem C. String reduction

There is a string of characters **a** and **b** with the length of no more than 255 characters. You can perform *the substring reduction* on the initial string in the following way: a substring **«a*a»** or **«b*b»** (where *** (asterisk)** denotes any character) can be reduces to a substring «*****» .

The task is to achieve a string of minimal possible length after several substring reductions.

Input data is the text file STRING.IN with the initial string.

Output data are located in a text file STRING.OUT containing a single line with the minimal possible length.

An example of the input data

aab

An example of the output data

3

Problem D. Add a queen

K chess queens (0 £ **K** < **M·N**) are located on a rectangular chessboard of size **M** x **N** (1 £ **M** , **N** £ 26) and there is no more than one queen in each cell. We consider a vacant cell is *under attack* if it's possible to move into it at least one of the queens with just one move according to the chess rules.

The task is to add one more queen into one of the vacant chessboard cells so that the number of cells under action will be minimal .

The chessboard rows are labeled with lowercase Latin letters starting with **a** from the bottom; the columns are labeled with numbers starting with 1 from the left. Thus the cells can be labeled as **a1, f23,** etc.

Input data is the text file QUEEN.IN which consists of **K** +1 lines: the first line contains the values **M, N** and **K** separated by one or several spaces; the next lines contain the labels of the cells where the queens are placed initially (one label per line).

Output data are located in a text file QUEEN.OUT which contains two lines. The first line contains the label of the cell where the queen should be places. (If there are several possible answers, pick the cell that is first in the lexicographic ordering of the cell labels). The second one contains the number of empty cells not under attack after the queen has been added.

An example of the input data

4 4 2

a1

a2

An example of the output data

c1

2

Problem E. "Bloody Mary "

There is a mixture of water, alcohol and tomato juice in three vessels. The volume ratio of components in the first, second and third vessels is equal $\mathbf{a_1 : a_2 : a_3}$, $\mathbf{b_1 : b_2 : b_3}$ and $\mathbf{c_1 : c_2 : c_3}$ correspondingly.

In what volume ratio is it necessary to take liquids from these vessels in order to create a mixture with a component ratio $\mathbf{d_1 : d_2 : d_3}$?

Input data: the text file MARY.IN consists of four lines. The first line contains values of parameters $\mathbf{a_1, a_2, a_3,}$ the others – values of parameters $\mathbf{b_1, b_2, b_3; c_1, c_2, c_3}$ and $\mathbf{d_1, d_2, d_3}$ correspondingly. All numbers are integers from 0 to 1000; each line contains at least one nonzero value. The values on each line are separated by colons (without spaces).

Output data are located in a text file MARY.OUT containing a single line with the result. The result has the same format as the input data.

If it is impossible to create necessary mixture, the output file should contain a single line with word *'impossible'*.

An example of the input data

1:0:0

0:1:0

0:0:1

3:2:3

An example of the output data

3:2:3

Problem F. Bridge over a rough river

A group of **N** travelers (1 £ **N** £ 50) has approached an old and shabby bridge and wishes to cross the river as soon as possible. However, there can be no more than two persons on the bridge at a time. Besides it's necessary to light the way with a torch for safe crossing but the group has only one torch.

Each traveler needs $\mathbf{t_i}$ seconds to cross the river on the bridge; **i**=1, …, **N** ($\mathbf{t_i}$ are integers from 1 to 100). If two travelers are crossing together their crossing time is the time of the slowest traveler.

The task is to determine minimal crossing time for the whole group.

Input data: the text file BRIDGE.IN consists of two lines: the first line contains the value of **N** and the second one contains the values of $\mathbf{t_i}$ (separated by one or several spaces) .

Output data are located in a text file BRIDGE.OUT which contains one line with the result.

An example of the input data

4

6 7 6 5

An example of the output data

29

Problem G. Convex hull

There are **M** circles ($0 \le \mathbf{M} \le 100$) on a plane. Some circles can have zero radius.

The task is to calculate the area of minimum convex domain containing all the circles.

Input data : the text file CIRCLE.IN contains the description of one test including:

· a line with value **M**;

· **M** lines; every line contains coordinates of the center and the radius for one circle. These are real numbers separated by spaces.

Output data are located in a text file CIRCLE.OUT containing a single line with the result given with accuracy 10^{-4}.

An example of the input data

4

0 0 1

0 4 1

4 4 1

4 0 1

An example of the output data

35.1416

Problem H. Last digit

Determine the last nonzero digit in value of expression

$$C_n^m = \frac{n!}{m!(n-m)!}$$

Input data: text file CNM.IN contains a single line with **n** and **m** separated by one or several spaces; **n, m** are natural numbers from 1 to 1000000, **n ³ m** .

Output data are located in a text file CNM.OUT containing a single line with the last nonzero digit.

An example of the input data

4 2

An example of the input data

6