

## Задача А. Electricity

Имя входного файла: `el.in`  
 Имя выходного файла: `el.out`  
 Ограничение по времени: 9 seconds  
 Ограничение по памяти: 64 Megabytes

You are given a symmetrical non-negative  $N \times N$  matrix  $A$  with zeroes on diagonal, i.e.  $a_{ij} = a_{ji} \geq 0$  and  $a_{ii} = 0$  for all  $i$  and  $j$ .

You are to find such symmetrical non-negative  $N \times N$  matrix  $B$  (i.e.  $b_{ij} = b_{ji} \geq 0$ ,  $b_{ii} = 0$  for all  $i$  and  $j$ ) that for all  $i, j$  and  $k$   $a_{ij} \geq b_{ij}$  and  $b_{ij} + b_{ik} \geq b_{kj}$ .

If there are several such matrices, you must find the matrix with the greatest sum of all  $b_{ij}$ .

### Формат входного файла

Input will consist of several test cases. Each test case will begin with one integer  $N$  ( $1 \leq N \leq 300$ ). Next  $N$  lines will have  $N$  integers each:  $j$ -th number in  $i$ -th line will be  $a_{ij}$ . All numbers will not exceed 1 000 000. The test with  $N = 0$  and the rest of the input file after this test must not be processed.

### Формат выходного файла

For each test case, output  $N + 1$  lines. On the first line, print one string: "Test case  $K$ :" (without quotes), where  $K$  is the number of test, and on the following  $N$  lines output the required matrix  $B$  in the same format as  $A$  in input, but without  $N$ .

### Пример

<code>el.in</code>	<code>el.out</code>
3	Test case 1:
0 1 3	0 1 2
1 0 1	1 0 1
3 1 0	2 1 0
0	

## Задача В. Nasty Mutant

Имя входного файла: `mutant.in`  
 Имя выходного файла: `mutant.out`  
 Ограничение по времени: 2 seconds  
 Ограничение по памяти: 64 Megabytes

Two guys who fought at ACM contests for almost all their student years have evolved a new kind of business: the shortest path evaluation. So they have founded a new company ACM (Abandon Cruel Meters) to help people to walk less and do more every day of their life. At first everything was fine and the algorithm they wrote at the contests blindly brought them enormous profits. But one day they faced a very strange client. That client was a mutant so he could cross any road in quite different manners. For example, he could merely walk or run or crawl or fly or swim (swim through asphalt???), etc. First he selected the path that will allow him to cross it only walking in the shortest time. Next, he wants to know the one which would allow him to run it faster (what if he would prefer running after all!). Next, he will select the "best crawling" path, then best flying, then best swimming and so on. So, you are to help ACM guys to prove that their company is capable of helping even such unordinary clients. Oh, and he is also brain-damaged, so he does not currently remember where his home, his office, his girl lives (btw, she is also a mutant), etc: So he asked to find such a path for every pair of source/destination junctions on the city map. Some city roads are one-way, some are two-way. For ease we will consider all of them as one-way. Two-way roads will be given as two one-way roads in the opposite directions.

### Формат входного файла

The first line contains three integer numbers:  $N$  — the number of road junctions,  $M$  — the number of different roads, and  $K$  — the number of mutant motions types.  $0 \leq N, K \leq 40$ . There is no limit for  $M$ , except that it will allow reading all the roads and solve the problem with asymptotically suitable algorithm without exceeding the time limit.  $M$  lines follow. Each of these lines has the form  $V_1 V_2 X_1 X_2 \dots X_K$ .  $V_1$  is the source junction,  $V_2$  is the destination junction of the road described by this line,  $1 \leq V_1, V_2 \leq N$ .  $X_i$  is the cost of the road (i.e. travel time) using  $i$ 'th mutant motion type.  $0 \leq X_i \leq 1000000, 1 \leq i \leq K$ .

### Формат выходного файла

Your output should contain  $N_2$  lines, one for every source-destination pair in the source-major order, each line should contain integers  $V_1 V_2 Y_1 Y_2 \dots Y_K$  giving the cost of the best path (in terms of the selection procedure described above) using only motion of type  $i$  on all of its roads from  $V_1$  to  $V_2$ . If there is no path from  $V_1$  to  $V_2$ , output " $V_1 V_2$  forget it" instead (without quotes). Please note again that the order of output lines is fixed, it is source-major order like if you would scan a matrix left to right, row-by-row.

### Пример

mutant.in	mutant.out
3 3 2	1 1 0 0
1 2 1 2	1 2 1 2
2 3 1 2	1 3 2 3
1 3 2 3	2 1 forget it
	2 2 0 0
	2 3 1 2
	3 1 forget it
	3 2 forget it
	3 3 0 0

## Задача С. Chessboard

Имя входного файла: `chess.in`  
Имя выходного файла: `chess.out`  
Ограничение по времени: 2 seconds  
Ограничение по памяти: 64 megabytes

A king wishes to go for a walk on a square chessboard with the following conditions:

1. Any two successive cells of the path must be adjacent, i.e., share an edge or a corner (thus, a cell may have up to eight adjacent cells).
2. Each cell must be visited exactly once; the first and the last cells of the path must coincide (thus, the first cell of the path will be actually visited twice).
3. The path must have no self intersections (if we think of a path as a closed polyline with vertices at cells' centers).

Your task is to find the maximal possible length of a king's path (here we mean the length of the polyline, not the number of king's moves).

### Формат входного файла

The only line of the input file contains an integer  $N$  ( $1 \leq N \leq 300$ ), denoting the size of the chessboard.

### Формат выходного файла

The only line of the output file must contain the length of the king's tour with at least three digits after the decimal point. The cells have side 1.

### Пример

<code>chess.in</code>	<code>chess.out</code>
1	0.000
2	4.000
3	9.414

## Задача D. Matrix

Имя входного файла: `matrix.in`  
Имя выходного файла: `matrix.out`  
Ограничение по времени: 4 seconds  
Ограничение по памяти: 64 megabytes

It is necessary to arrange numbers from 0 to  $2^{(N+M)} - 1$  in the matrix with  $2^N$  rows and  $2^M$  columns. Moreover, numbers occupying two adjacent cells must differ only in single bit in binary notation. Cells are adjacent if they have common side. Matrix is cyclic, i.e. for each row the leftmost and rightmost matrix cells are considered to be adjacent (the topmost and the bottommost matrix cells are also adjacent).

### Формат входного файла

The first line of input contains two integers  $N$  and  $M$  ( $0 < N, M; N + M \leq 20$ ).

### Формат выходного файла

Output file must contain the required matrix in a form of  $2^N$  lines of  $2^M$  integers each.

### Пример

<code>matrix.in</code>	<code>matrix.out</code>
1 1	0 2 1 3

## Задача E. Sum of Divisors

Имя входного файла: `sumdiv.in`  
Имя выходного файла: `sumdiv.out`  
Ограничение по времени: 2 seconds  
Ограничение по памяти: 64 megabytes

Vasya is fond of Number Theory. So he is interested in different curious number-theoretic functions. Today he is interested in the function  $\sigma'(n)$  defined for integer  $n \geq 1$  as the sum of all positive divisors  $d$  of  $n$  which are not multiples of four:

$$\sigma'(n) = \sum_{\substack{d|n \\ d \bmod 4 \neq 0}} d$$

You are to compute the value of  $\sigma'(n)$  for a given  $n \leq 10^9$  since Vasya is too lazy for doing it himself.

### Формат входного файла

The only line of input contains exactly one integer number  $n$ .

### Формат выходного файла

Just output  $\sigma'(n)$ . If there are no such divisors  $d$ , output 0 instead.

### Пример

<code>sumdiv.in</code>	<code>sumdiv.out</code>
10	18
12	12
16	3

## Задача F. Printed PR

Имя входного файла: `pr.in`  
Имя выходного файла: `pr.out`  
Ограничение по времени: 2 seconds  
Ограничение по памяти: 64 megabytes

Small PR-agency got an order to print a very big lot of agitational materials. Agency should print and deliver  $N$  leaflets. Agency is very small and has only one printer, but it can print any leaflet. Leaflets can be different, so it is possible that times of printing of leaflets will differ. To print  $i$ -th leaflet the printer needs  $T_i$  minutes. When leaflet is printed, it should be immediately delivered to its addressee. The agency has unlimited number of couriers, so the courier gets the leaflet as soon as it printed and goes to the addressee. It takes  $L_i$  minutes to deliver  $i$ -th leaflet. You were hired by agency to calculate the minimal time required to finish the job (this is an interval of time from the beginning of printing to the moment when the last leaflet is delivered), considering that the leaflets can be printed in any order.

### Формат входного файла

The first line contains a number of leaflets — integer number  $N$  ( $1 \leq N \leq 100$ ). The second line contains  $N$  integer numbers  $T_i$  ( $1 \leq i \leq N$ ). Third line contains  $N$  integer numbers  $L_i$  ( $1 \leq i \leq N$ ). You can assume that  $1 \leq T_i, L_i \leq 1000$ .

### Формат выходного файла

You should output only one number — the answer for the problem.

### Пример

<code>pr.in</code>	<code>pr.out</code>
2	4
2 1	
2 1	