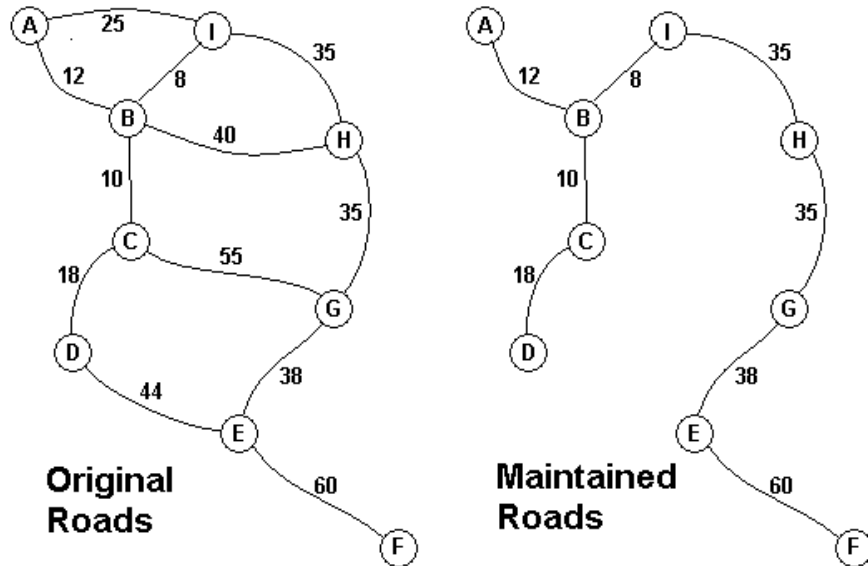


Jungle Roads

Task code: JUNGLE

Time for one test: 1 sec, memory limit: 1 Mb

The Head Elder of the tropical island of Lagrishan has a problem. A burst of foreign aid money was spent on extra roads between villages some years ago. But the jungle overtakes roads relentlessly, so the large road network is too expensive to maintain. The Council of Elders must choose to stop maintaining some roads. The map above on the left shows all the roads in use now and the cost in dollars per month to maintain them. Of course there needs to be some way to get between all the villages on maintained roads, even if the route is not as short as before. The Head Elder would like to tell the Council of Elders what would be the smallest amount they could spend in dollars per month to maintain roads that would connect all the villages. The villages are labeled A through I in the maps above. The map on the right shows the roads that could be maintained most cheaply, for 216 dollars per month. Your task is to write a program that will solve such problems.



The input file JUNGLE.IN contains one data set. A data set starts with a line containing only a number n , which is the number of villages, $1 < n < 27$, and the villages are labeled with the first n letters of the alphabet, capitalized. The first line is followed by $n-1$ lines that start with village labels in alphabetical order. There is no line for the last village. Each line for a village starts with the village label followed by a number, k , of roads from this village to villages with labels later in the alphabet. If k is greater than 0, the line continues with data for each of the k roads. The data for each road is the village label for the other end of the road followed by the monthly maintenance cost in dollars for the road. Maintenance costs will be positive integers less than 100. All data fields in the row are separated by single blanks. The road network will always allow travel between all the villages. The network will never have more than 75 roads. No village will have more than 15 roads going to other villages (before or after in the alphabet). In the sample input below, the first data set goes with the map above.

The output file JUNGLE.OUT contains one integer: the minimum cost in dollars per month to maintain a road system that connect all the villages.

Example of input data

```

9
A 2 B 12 I 25
B 3 C 10 H 40 I 8
C 2 D 18 G 55
D 1 E 44
E 2 F 60 G 38
F 0
G 1 H 35
H 1 I 35
    
```

Example of output data

216

Tanning Salon

Task code: SALON

Time for one test: 1 sec, memory limit: 1 Mb

Tan Your Hide, Inc., owns several coin-operated tanning salons. Research has shown that if a customer arrives and there are no beds available, the customer will turn around and leave, thus costing the company a sale. Your task is to write a program that tells the company how many customers left without tanning.

The input file SALON.IN consists of a single line containing a positive integer, representing the number of tanning beds in the salon, followed by a space, followed by a sequence of uppercase letters. Letters in the sequence occur in pairs. The first occurrence indicates the arrival of a customer, the second indicates the departure of that same customer. No letter will occur in more than one pair. Customers who leave without tanning always depart before customers who are currently tanning. There are at most 20 beds per salon.

The output file SALON.OUT contains one integer telling how many customers, if any, walked away. If all customers tan successfully, the file contains 0.

Example of input data

3 GACCBDDBAGEE

Example of output data

1

Three operations

Task code: W_E_R

Time for one test: 1 sec, memory limit: 16 Mb

There are two strings: S_1 , of length k ($3 \leq k \leq 100$), and S_2 that is initially empty. The following operations are allowed:

- W : write the first character of string S_1 to the end of string S_2 ;
- E : erase the first character of S_1 ;
- R : rotate S_1 so that the first character takes the last place, the second character takes a place before the last ..., and the last character becomes the first.

You need to find a sequence of operations that writes a word T to string S_2 . The length of the word T is less than 11 characters. The length of the sequence must not exceed 150. It's always possible to find the desired sequence.

The input file W_E_R.IN consists of two strings: S_1 , and T .

The output file W_E_R.OUT contains a sequence of operations separated by a single space.

Example of input dataNOGETISEOROSILE
NOSOROGExample of output data

W E W R E E E W E W E W R W E W

Robots

Task code: ROBOTS

Time for one test: 1 sec, memory limit: 1 Mb

Your company provides robots that can be used to pick up litter from fields after sporting events and concerts. Before robots are assigned to a job, an aerial photograph of the field is marked with a grid. Each location in the grid that contains garbage is marked. All robots begin in the Northwest corner and end their movement in the Southeast corner. A robot can only move in two directions, either to the East or South. Upon entering a cell that contains garbage, the robot will pick it up before proceeding. Once a robot reaches its destination at the Southeast corner it cannot be repositioned or reused. Since your expenses are directly proportional to the number of robots used for a particular job, you are interested in finding the minimum number of robots that can clean a given field. For example, consider the field map shown in Figure 1 with rows and columns numbered as shown and garbage locations marked with a 'G'. In this scheme, all robots will begin in location 1,1 and end in location 6, 7.

	1	2	3	4	5	6	7
1		G		G			
2				G		G	
3							
4				G			G
5							
6						G	

Figure 1 - A Field Map

Figure 2 below shows two possible solutions, the second of which is preferable since it uses two robots rather than three.

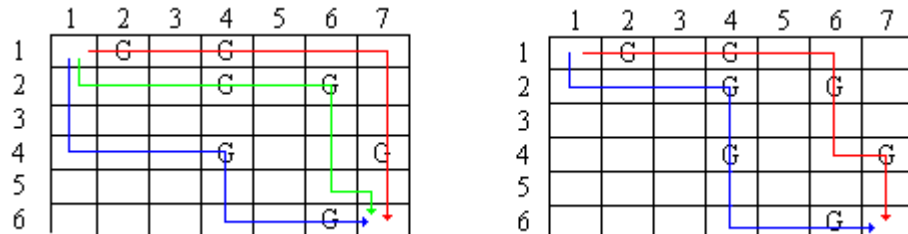


Figure 2 - Two Possible Solutions

Your task is to create a program that will determine the minimum number of robots needed to pick up all the garbage from a field.

The input file ROBOTS.IN contains a description of a field map. The first line of the file contains one integer n: a number of garbage locations. The next n lines contain one garbage location each. A garbage location consists of two integers, the row and column, separated by a single space. The rows and columns are numbered as shown in Figure 1. The garbage locations will be given in row-major order. No single field map will have more than 24 rows and 24 columns. The sample input below shows the field map from Figure 1.

The output file ROBOTS.OUT will consist of a single line containing the minimum number of robots needed to clean the field.

Example of input data

```
7
1 2
1 4
2 4
2 6
4 4
4 7
6 6
```

Example of output data

```
2
```

Изношенный механизм

Task code: REMOTE

Time for one test: 1 sec, memory limit: 16 Mb

Пульт управления телевизором имеет N кнопок ($3 \leq N \leq 20$) для включения одного из N принимаемых каналов. Когда пульт был новым, нажатие любой из этих кнопок автоматически приводило к отжатию остальных, так что в каждый момент времени только одна из кнопок переключения каналов была включена. К сожалению, с течением времени механизм пульта изнашивался, и некоторые кнопки не отключаются при нажатии других кнопок. Таким образом, одновременно могут быть включены несколько каналов. Выбор нужной программы становится непростой задачей! С другой стороны, финансовые проблемы не позволяют Вам купить новый телевизор или хотя бы отремонтировать пульт...

После долгих манипуляций с пультом Вы определили, какие кнопки отжимаются при нажатии кнопки с номером k ($1 \leq k \leq N$) и приноровились включать нужный канал нажатием последовательности из нескольких кнопок.

Вам требуется определить минимальное количество нажатий кнопок переключения каналов, необходимых для включения канала с номером k (т.е. для перевода пульта в такое состояние, при котором кнопка k нажата, а все остальные – отжаты). Следует помнить, что повторное нажатие уже нажатой кнопки не вызывает никакого эффекта.

Входные данные находятся в текстовом файле REMOTE.IN, и содержат $N+2$ строки. Первая строка этого файла содержит значения N и k . Во второй строке записано текущее состояние пульта в виде последовательности из N нулей и единиц, где 1 соответствует нажатой кнопке, а 0 – отжатой. Наконец, каждая из последующих N строк содержит количество кнопок, отжимающихся при нажатии кнопки с номером k ($1 \leq k \leq N$), и их номера (нумерация кнопок начинается с единицы). Числа в строках входного файла разделяются одним или несколькими пробелами. Тесты подобраны таким образом, что задача всегда имеет хотя бы одно решение.

Выходные данные помещаются в текстовый файл REMOTE.OUT. Единственная строка этого файла содержит искомое число нажатий кнопок.

<u>Example of input data</u>	<u>Example of output data</u>
5 3	3
1 1 0 0 1	
4 2 3 4 5	
4 1 3 4 5	
2 2 4	
0	
4 1 2 3 4	

Натуральный ряд чисел

Task code: NATURAL

Time for one test: 3 sec, memory limit: 1 Mb

Представления чисел натурального ряда в системе счисления с основанием k ($2 \leq k \leq 36$) записаны в строку без каких-либо разделителей, так что начало этой строки при $k = 10$ имеет вид

123456789101112131415161718192021 ...

(Для записи цифр в системе счисления с основанием k используются первые k символов из последовательности '0', '1', ..., '9', 'A', ..., 'Z'.)

Вам требуется определить первое вхождение десятичной записи числа N ($1 \leq N \leq 10^4$) в этой строке.

Входные данные находятся в текстовом файле NATURAL.IN, и содержат единственную строку со значениями N и k , разделенными одним или несколькими пробелами.

Выходные данные помещаются в текстовый файл NATURAL.OUT. Единственная строка этого файла содержит позицию, начиная с которой в строке записано первое десятичное представление исходного числа. Нумерация позиций начинается с единицы. Если решение получить невозможно, выходной файл должен содержать значение 0. Входные и выходные данные представлены в десятичной системе счисления!

<u>Example of input data</u>	<u>Example of output data</u>
142 10	73

Склеивающиеся точки

Task code: POINTS Time for one test: 1 sec, memory limit: 16 Mb.

На плоскости расположены N различных точек ($0 \leq N \leq 5000$). Координаты каждой точки – целые числа в интервале от 1 до 1000. Две точки $A(x_1, y_1)$ и $B(x_2, y_2)$ из некоторой последовательности, в которой A находится непосредственно перед B , будем считать склеивающимися, если $x_1 = y_2$.

Вам требуется написать программу, которая генерирует такую последовательность исходного множества точек, что любые две соседние точки в этой последовательности можно склеить.

Входные данные находятся в текстовом файле POINTS.IN, содержащем $N+1$ строку. Первая строка этого файла содержит количество точек, а каждая из последующих строк описывает одну точку и содержит ее координаты, разделенные одним или несколькими пробелами.

Тесты подобраны таким образом, что задача всегда имеет хотя бы одно решение.

Выходные данные помещаются в текстовый файл POINTS.OUT, содержащий N строк. В i -ой строке содержатся координаты i -ой точки последовательности, разделенные одним пробелом. Если N равно 0, то программа должна создать пустой файл.

В случае, когда задача допускает несколько решений, достаточно найти любое из них.

<u>Example of input data</u>	<u>Example of output data</u>
5	4 7
7 6	9 4
4 7	8 9
8 9	6 8
9 4	7 6
6 8	

Экипажи

Task code: CREW Time for one test: 1 sec, memory limit: 16 Mb.

Авиакомпания, в которой Вы работаете менеджером по кадрам, должна сформировать N экипажей самолетов ($1 \leq N \leq 5000$). Каждый экипаж состоит, помимо других должностей, из двух пилотов: командира и второго пилота. Не допускается, чтобы командир экипажа имел меньший налет часов, чем его второй пилот.

В Вашем распоряжении имеется $2*N$ кандидатур пилотов, налет часов каждого пилота известен. При заключении контракта с каждым пилотом была оговорена его зарплата в качестве командира и в качестве второго пилота. Первая величина, естественно, не меньше второй...

Вам необходимо сформировать экипажи таким образом, чтобы минимизировать суммарный фонд зарплаты пилотов. При этом допускается, чтобы зарплата второго пилота была большей, чем зарплата командира его экипажа.

Входные данные находятся в текстовом файле CREW.IN, содержащем $2*N+1$ строку. Первая строка этого файла содержит количество пилотов, каждая из последующих строк соответствует одному пилоту и содержит его зарплату в должности капитана и в должности второго пилота. Строки с описаниями зарплат строго упорядочены по возрастанию налета часов каждого пилота.

Выходные данные помещаются в текстовый файл CREW.OUT. Единственная строка этого файла содержит одно число – искомый фонд заработной платы пилотов.

<u>Example of input data</u>	<u>Example of output data</u>
6	31
5 1	
9 8	
6 2	
10 9	
5 3	
6 1	