



# ACM International Collegiate Programming Contest

## 2007

South American Regional Contests

*November 9-10, 2007*

### Warmup Session

*This problem set contains 2 problems; pages are numbered from 1 to 3.*

This problem set is used in simultaneous contests hosted in the following countries:

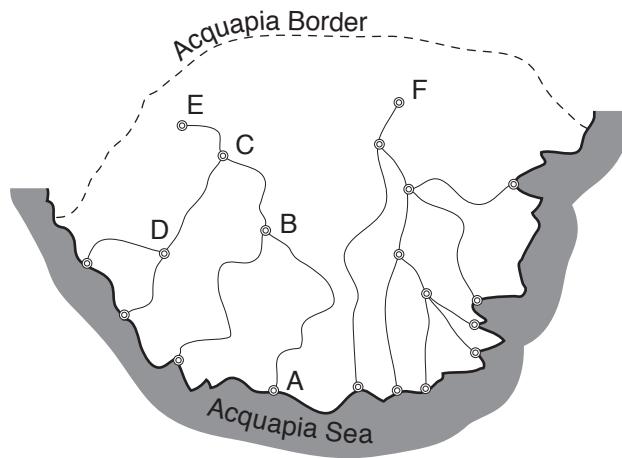
- Argentina
- Bolivia
- Brazil
- Chile
- Colombia
- Peru
- Venezuela

# Problem A

## Acquapia

*Source file name: acqua.c, acqua.cpp or acqua.java*

Acquapia is a small country crossed by several rivers, all of them navigable. All rivers have their sources inside Acquapia, in the mountains, and all rivers end at sea – a river never ends at another river. During its course (not at its source) a river may split into two or more streams. There is a city wherever a river begins, splits or ends, and each city touches at most one river. The figure below shows a map of Acquapia, with two rivers and twenty cities. For clarity, only some cities are labeled. In the figure, *E* and *F* are cities that have river sources; *B*, *C* and *D* are cities where a river splits into different streams; and *A* is a city where a river ends.



Being so widespread, rivers are the main means of transportation in Acquapia. But since the country is at war with a neighboring country, it is not safe for ships to cross the sea, so ships can only navigate in rivers. That means that to go from city *A* to city *D*, a ship must travel upstream (against the river flow) from *A* to *B* and *C*, make a *stream change* at *C* and then travel downstream from *C* to *D*. *Stream change*, which is the operation of changing from navigating upstream to navigating downstream, is difficult and somewhat dangerous, since river streams are strong in Acquapia, and the turns are sharp. Notice that to navigate from *B* to *E* (or from *E* to *B*) no stream change is necessary. Also, due to the war, it is not possible to navigate, for example, from *B* to *F*.

Acquapia Cargo Management (ACM) is a software house that specializes in systems for navigation. ACM wants you to write a program that, given the description of rivers, the description of cities, and a set of queries, each composed by a pair of cities *X* and *Y*, answer the following questions:

- “now that ships cannot cruise the sea, is it possible to navigate from city *X* to city *Y*”?
- “if it is possible to navigate from *X* to *Y*, does a ship need to make a stream change? If so, in which city, so that the distance travelled is the shortest possible?”

## Input

The input contains several test cases. The first line of a test case contains four integers  $C$ ,  $R$ ,  $S$  and  $Q$ , separated by single spaces, representing respectively the number of cities ( $2 \leq C \leq 10^3$ ), the number of rivers ( $1 \leq R \leq C/2$ ), the number or river sections ( $1 \leq S \leq C - 1$ ) and the number of queries that will be made ( $1 \leq Q \leq 2 \times 10^5$ ). Cities are numbered sequentially from 1 to  $C$ . The second line of a test case contains  $R$  distinct integers, separated by single spaces, indicating the cities that are river sources. Each of the next  $S$  lines describes a different river section and contains two integers  $X$  and  $Y$ , separated by a single space, indicating that there is a section of river that flows from city  $X$  to city  $Y$  ( $X \neq Y$ ). Then comes  $Q$  lines, each describing a query. Each query is composed of two integers  $A$  and  $B$ , separated by a single space, representing two cities ( $A \neq B$ ). The end of input is indicated by  $C = R = S = Q = 0$ .

*The input must be read from standard input.*

## Output

For each test case your program must output  $Q + 1$  lines, where  $Q$  is the number of queries in the test case. The first line must be empty; each of the following  $Q$  lines must contain one integer, representing the answer to the corresponding query, in the same order as the input. If it is not possible to navigate between the two cities in the query, print ‘-1’. If it is possible to navigate between the two cities but no stream change is necessary, print ‘0’. Otherwise, print the integer that represents the city where the stream change must be made.

*The output must be written to standard output.*

Sample input	Output for the sample input
4 1 3 3	
1	2
1 2	0
2 3	0
2 4	
4 3	0
2 1	-1
1 3	3
7 2 5 3	
1 6	
1 3	
3 2	
3 5	
3 4	
6 7	
6 7	
1 7	
5 4	
0 0 0 0	

# Problem B

## Og

*Source file name: og.c, og.cpp or og.java*

Og is a caveman with many children, and he wants to count them all. Og counts his sons with his left hand and his daughters with his right hand.

However, Og is very dumb, and can't add the two counts, so he asked you to write him a program that will do the addition.

### Input

The input contains several test cases. Each test case consists of a single line containing two integers  $L$  and  $R$ , separated by a single space, indicating respectively the number of sons and daughters ( $1 \leq L, R \leq 5$ ). The end of input is indicated by  $L = R = 0$ .

*The input must be read from standard input.*

### Output

For each test case in the input print a line containing a single integer indicating how many children Og has.

*The output must be written to standard output.*

Sample input	Output for the sample input
2 2	4
2 3	5
5 5	10
1 1	2
0 0	