

# Relational MapReduce Patterns #4 - Unions

## Mappers and Reducers

[Here's](#) a quick but comprehensive introduction to the idea of splitting tasks into a MapReduce model. The four important functions involved are:

```
Map (the mapper function)
EmitIntermediate(the intermediate key,value pairs emitted by the mapper functions)
Reduce (the reducer function)
Emit (the final output, after summarization from the Reduce functions)
```

We provide you with a single system, single thread version of a basic MapReduce implementation.

## Task

Given two sets of integers, **R** and **S** select and display the union.

The code for the MapReduce class, reading and splitting the text, parts related to IO etc. has already been provided. However certain parts of the mapper and reducer functions are incomplete. You need to replace the questionmarks (?). Your task is to fill up these question marks appropriately, such that the program works and accomplishes the specified task. Also, this program may output certain information to the error stream. This information has been logged to help beginners gain a better understanding of the the intermediate steps in a map-reduce process.

## Input Format

The first line contains two space separated integers **Nr** and **Ns**, which are the number of elements in the set **R** and **S** respectively. This is followed by **Nr** integers, the elements of set **R**, each on a new line, such that - 100 <= **X** <= 100. This is followed by **Ns** integers, the elements of set **S**, each on a new line, such that - 100 <= **X** <= 100. Also, 10 <= **Nr**,\***Ns**\* <= 100

## Output Format

Output each of the integers in the union of the sets **R** and **S**. The relative ordering of integers in the union set should be same as that in set **R** and **S** with **R** having higher priority.

## Sample Input

```
10 10
-51
-43
74
-96
24
-14
11
77
-45
-90
45
8
29
0
-43
```

```
-13
-72
71
-96
-26
```

## Sample Output

```
-51
-43
74
-96
24
-14
11
77
-45
-90
45
8
29
0
-13
-72
71
-26
```

## Explanation

We list out all the elements present in the union of **R** and **S**.