Queen Mary
University of London

Department of Computer Science

1st DCS Programming Competition

30th Jan 2008

Problem 1 – Hello \<input>

(Your program should be named "Hello.java")

Problem description: Your first task is simple: Write a parametrised "Hello World" program.

Input: The input consists of a number N, followed by N lines each containing a word.

Output: Your task is to output for each input word X the sentence "Hello X" on a new line.

Sample Input

```
3
World
Mum
Everybody
```

Sample Output

```
Hello World
Hello Mum
Hello Everybody
```

Problem 2 – Sorting Things Out

(Your program should be named "Sorting.java")

Problem description: This is another simple problem for everybody to solve. You are asked to write a program that sorts any given sequence of integers, removing duplicates.

Input: The input consists of several lines. Each line starts with a number N ($0 \leq N \leq 10000$) indicating the size of the list that follows. A value N = 0 indicates the end of the input. The numbers you are asked to sort are in the range [1...5000].

Output: Your task is to return the list in increasing order and without repetitions.

Sample Input

```
6 3 5 9 2 2 10
10 8 8 2 8 6 4657 4 3 5 1
0
```

Sample Output

```
2 3 5 9 10
1 2 3 4 5 6 8 4657
```

Problem 3 – Simply Subsets[1]

(*Your program should be named "Subsets.java"*)

Problem description: After graduating from Queen Mary University, you obtained a job at Top Shelf Software, Inc., as an entry-level computer engineer. On the first day, your manager sits down with you and tasks you with the following job: ``As an evaluation for all of our new hires, we require them to write a program to determine the relationship between pairs of sets. I'm quite sure that you'll do well; my confidence is high. Here's a list of requirements for what the program should do. Good luck.''

Input: Your program should accept an even number of lines of text. Each pair of lines will represent two sets; the first line represents set A, the second line represents set B. Each set will be given as list of at most 100 distinct (not necessarily sorted) positive integers in the range [1…5000]. A zero (0) indicates the end of the list. The zero (0) does not need to be considered part of the set. For example, the list "55 27 0" represents the set {55, 27}. Note that sets might be empty. Two empty sets indicate the end of the input.

Output: After each pair of lines has been read in, the sets should be compared and one of the following responses should be output:

`A is a proper subset of B` – If every element of A also appears in B
`B is a proper subset of A` – If every element of B also appears in A
`A equals B` – If the two sets contain the same elements
`A and B are disjoint` – If A and B do not contain any common element
`I'm confused!` – If none of the above holds

Sample Input

```
55 27 0
55 27 0
9 1995 24 0
9 24 0
1 2 0
2 3 0
0
0
```

Sample Output

```
A equals B
B is a proper subset of A
I'm confused!
```

---

[1] From acm.uva.es/p/v4/496.html

Problem 4 – Self-describing Sequence[2]

(Your program should be named "Sequence.java")

Problem description: Solomon Golomb's self-describing sequence is the only non-decreasing sequence of positive integers with the property that it contains exactly *f(n)* occurrences of *n* for each *n*. A few moments thought reveals that the sequence must begin as follows:

n	1	2	3	4	5	6	7	8	9	10	11	12
$f(n)$	1	2	2	3	3	4	4	4	5	5	5	6

In this problem you are expected to write a program that calculates the value of *f(n)* given the value of *n*.

Input: The input may contain multiple test cases. Each test case occupies a separate line and contains an integer *n* ($1 \le n \le 200000$). The input terminates with a test case containing a value *0* for *n* and this case must not be processed.

Output: For each test case in the input your program should output the value of *f(n)* on a separate line.

Sample Input

```
1
9
100
9999
123456
0
```

Sample Output

```
1
5
21
356
1684
```

---

[2] By Miguel Revilla, 2000-12-26 (http://acm.uva.es/problemset/v100/10049.html)

Problem 5 – Take the Land[3]

(Your program should be named "Land.java")

Problem description: The poor man went to the King and said, "Lord, I cannot maintain my family. Please give me some wealth so that I can survive with my wife and children." The King replied, "I shall grant you a piece of land so that you can cultivate and grow food for your family. In the southern part of the Kingdom there is a rectangular forest. Trees have been planted there at regular intervals. Some of the trees have been cut for use. You are allowed to take any rectangular piece of land that does not contain any tree. You need not go to the forest to select the piece of land. I have a map containing 1s at places where there is a tree and 0s at points where the tree has been cut."

Help the poor man to find out the largest piece of land. Area of the land is measured in units of number of trees that were there. Your program should take a matrix of 1s and 0s as input and output the area of the largest rectangular piece of land that contain no tree.

Input: The input file may contain multiple test cases. The first line of each test case contains two integers M and N $(1 \leq M, N \leq 100)$ giving the number of rows and columns in the matrix that follows. Each of the next M lines contains N symbols (either 0 or 1). Two consecutive symbols in a line will be separated by a single space. The input terminates with two zeros for M and N.

Output: For each test case in the input print a line giving the area (in terms of the number of trees that were there) of the largest rectangular piece of land containing no tree.

Sample Input

```
2 4
0 1 1 1
0 1 0 1
6 7
0 1 1 0 1 1 0
0 0 0 0 0 1 0
1 0 0 0 0 0 1
0 1 0 0 0 0 1
1 1 0 0 0 1 0
1 1 0 1 1 0 0
0 0
```

Sample Output

```
2
12
```

---

[3] By Rezaul Alam Chowdhury (http://acm.uva.es/p/v100/10074.html)

Problem 6 – Vito's family[4]

(Your program should be named "Vito.java")

Problem description: The world-known gangster Vito Deadstone is moving to New York. He has a very big family there, all of them living in Lamafia Avenue. Since he will visit all his relatives very often, he is trying to find a house close to them. Vito wants to minimize the total distance to all of them and has blackmailed you to write a program that solves his problem.

Input: The input consists of several test cases. The first line contains the number of test cases. For each test case you will be given the integer number of relatives r ($1 \leq r \leq 500$) and the street numbers (also integers) $s_1, s_2, ..., s_i, ..., s_r$ where they live ($1 \leq s_i \leq 30000$). Note that several relatives could live in the same street number.

Output: For each test case your program must write the minimal sum of distances from the optimal Vito's house to each one of his relatives. The distance between two street numbers s_i and s_j is $d_{ij} = |s_i - s_j|$.

Sample Input

```
2
2 2 4
3 2 4 6
```

Sample Output

```
2
4
```

---

[4] By Miguel Revilla, 2000-11-19 (http://acm.uva.es/problemset/v100/10041.html)