TAMS Tournament 2013
Computer Science Examination

TAMS Tounament

November 9th 2013

Welcome to the Computer Science competition of the 2013 TAMS Tournament! This competition consists of multiple programming exercises. You will have access to one computer per student.

Solutions must be written in C, C++, Java, and Python. You may work on problems in any order until time is called. When time expires please surrender your computer to the panal of judges seated at the front of the room. Please write the question problem at the beginning of the code. Please have a copy of the live code and raw code open on your computer when you turn in your computer. Judges will then run seperate test set to verify run time.

Each problem will consist of a problem description, followed by sample input and output.Please ensure that your program produces the sample output when given the sample input before submission.If you believe a problem description is ambiguous, please request clarification from the judges.

Different problems are worth different numbers of points. Problems with higher point totals are more difficult.

0 points will be awarded for programs that:

- Do not give an output when given the correct input
- Do not follow input criteria
- Returns an error message

Partial credit may be awarded for programs that

- Gives the correct output but vary by slightly in formatting (i.e extra blank space, incorrect number of lines)
- Returns a partial output (i.e 2 out of 4 lines)

Full credit will be given for programs that take the correct input and produce the exact output.

Tie breakers will be determined by simplest code.

0 This is worth points? (1 point)

This problem is included to allow you to test your programming environment and become familiar with the submission system.

Your program will receive two positive integers, one per line, as inputs. It should output the sum of those two integers. Your output should end with a newline, and contain no space after the integer. All inputs and outputs will be between 0 and $2^31 - 1$.

0.1 Sample input

```
2
3
```

0.2 Sample output

```
5
```

1 Cycles (1 point)

Let the function

$$C(n) = \begin{cases} n/2 & n \text{ is even} \\ 3n+1 & n \text{ is odd} \end{cases}$$

The cycle length of n is the number of iterations of $C(n)$ needed to reach an output of 1, including the final iteration. For example, the outputs of repeated calls to $C(n)$, where $n = 5$ initially, are

$$16, 8, 4, 2, 1$$

giving a cycle length of 5. The cycle length of 1 is 0.

For this problem, you will be given pairs of integers a and b, where $1 < a < b < 2^{16}$. For each pair, find the integer between a and b (including a and b) that has the greatest cycle length, and print that length.

1.1 Input

The input will consist of multiple lines of integer pairs. On each line, the two integers will be separated by a single space. There will be no more than 20 testcases. The input will be terminated by the line `0 0`

1.2 Output

For each line of input, the program should output the maximum cycle length between the two input numbers, which is found as described above. The answer should be on a line by itself, and be followed by a newline. Do not produce any output for the `0 0` terminator.

1.3 Sample input

```
1 10
100 200
19 24
1 27
1000 2000
0 0
```

1.4 Sample output

```
19
124
20
111
81
```

2 String Matching (2 points)

Given a pattern and a string, find all the locations (in increasing order, 0 indexed) in the string (0 indexed) where the pattern occurs. If the pattern contains multiple characters, find the indeces (within the string) of the first character in the match.

2.1 Input

The input will begin with a line containing the pattern, which will consist of uppercase ASCII letters and spaces. The pattern is gauranteed to be no longer than the string. The next line will contain the string, which will also contain only uppercase ASCII letters and spaces. The trailing newlines in the pattern and string should be discarded.

2.2 Output

The output consists of a series of integers, one per line, that are indexas of ste string where a match occurs, one per line.

2.3 Sample input

```
THE
THE WORD THE OCCURS THREE TIMES IN THE INPUT
```

2.4 Sample output

```
0
9
35
```

3 Coinage (2 points)

A country issues 1, 2, 5, 10, 25, and 50 cent coins. Givin an amount of change $0 < c \leq 5000$ cents, find the minimum number of coins needed to give out that amount of change, and how much of each coin type you would need.

3.1 Input

The input will consist of multiple (no more than 20) testcases, one per line. Each testcase will contain an integer c, the number of cents you must distribute as change on a line by itself.

The last line of input will contain only 0, signifying that there are no more testcases.

3.2 Output

For each testcase, output one line containing the number of each denomenation of coin you would need to produce that amount of change in order of decreasing denomenation, separated by spaces. That is, output the number of 50 cent coins, then the number of 25 cent coints, and so on. If you do not need a particular type of coin, output zero.

3.3 Sample input

```
78
94
4999
69
0
```

3.4 Sample output

```
1 1 0 0 1 1
1 1 1 1 2 0
99 1 2 0 2 0
1 0 1 1 2 0
```

4 Longest Common Substring (3 points)

Given two strings A and B, find the longest string C that is a substring of both A and B. If A and B have no substrings in common, print `No match`. If there more than one longest substring, print `Multiple matches`. If the longest substring appears multiple times in one of the strings, it is considered to be a multiple match.

4.1 Input

The input will consist of two strings of no more than 64 uppercase ASCII letters, one per line.

4.2 Output

The output should contain the longest common substring of A and B on a line by itself. If the two strings have no common substring, your program should output `No match`. If there are multiple candidates for the longest common substring, or if the longest substring appears multiple times, you should print `Multiple matches`.

4.3 Sample input

```
BABA
ABAD
```

4.4 Sample output

```
ABA
```

5 Parakeets (2 or 3 points)

A group of $N(3 \leq N \leq 25)$ parakeets are keeping the other animals in the zoo awake at night. The parakeets are kept in an 1000 inch by 1000 inch cage. The parakeets like to roost on the top level of the cage at night, but they are spread out over the top level. Each parakeet squawks loudly enough so that every other parakeet in the cage can hear it. Therefore, the loudness of a parakeet is the distance between that parakeet and the farthest parakeet from it. Given the coordinates of the parakeets, find the loudness of the cage, which is the sum of the loudness of all the parakeets.

5.1 Input

Line 1 will contain an integer N between 3 and 25 inclusive, the number of parakeets in the cage. Lines 2 to $N + 1$ contain two integers between 0 and 1000 inclusive, the x and y coordinates of the parakeets.

5.2 Output

The total loudness of the cage rounded to the nearest integer.

5.3 Sample input

```
4
3 1
6 5
2 4
5 3
```

5.4 Sample output

```
17
```

6 Elephant

Krzysztof has killed an elephant. He is very hungry, but can only eat in chunks of 6 lb, 11 lb, and 13 lb. The entire elephant's weight w is given. Krzysztof is also very environmentally conscious, and doesn't want to have any elephant left over. Can Krzysztof eat the entire elephant without wasting any, and if so, how many chunks will he have to eat?

6.1 Input

The first line is the number of inputs. Each input will be a single integer, the weight of the elephant.

6.2 Output

The output should read `Yes` if Krzysztof can eat the elephant followed by the minimum number of chunks he must eat, and `No` if he can't.

6.3 Sample input

```
2
75
27
```

6.4 Sample output

```
Yes 7
No
```

7 Egyptian Fractions

The Egyptians, as advanced as they were because of the ancient aliens and Lord Xenu and Giorgio A. Tsoukalos, were pretty bad at fractions. They would express every fraction as the sum of unit fractions (1 for the numerator). Given two integers aa and b, find the set of n's so that $\frac{a}{b} = \frac{1}{n_1} + \frac{1}{n_2} \ldots$

7.1 Input

The first line contains the number of testcases. For each testcase, two integers a and b will be given on a line separated by spaces.

7.2 Output

For each testcase, output the values of n sorted from smallest to largest on one line separated by spaces.

7.3 Sample input

```
4
4 5
6 7
9 5
4 6
```

7.4 Sample output

```
2 4 20
2 3 42
1 2 4 20
2 6
```