CSC4000W ~ Theory of Algorithms Assignment 1 ~ 2017

Submission

Submit your source file within a single compressed, '.ZIP', archive.

NB: create a '.ZIP' archive, not a gzipped, '.gz', or tar-gzipped, '.tgz', or other kind of file.

Make sure your source file is the only item within the archive. Especially, avoid submitting an archive containing a folder containing the file.

When submitting a Java source file copied from an editor like Eclipse or Netbeans, please remove any package line that may appear at the beginning of the code.

Question 1 [50 marks]

You are working as a consultant for farmers that have mixed herds of cattle, sheep and goats. Each farmer would like to maximise the profit her herd produces. In a given year, cattle produce profit p_c, sheep produce profit p_s and goats produce profit p_g. However, in a given year, the farmer's farm only produces an amount of food F and the cattle, sheep and goats eat an amount of food f_c, f_s and f_g, respectively, per individual animal. Moreover, in a given year, the farmer's farm only has an available amount of water W and the cattle, sheep and goats drink an amount of water w_c, w_s and w_g, respectively, per individual animal. Due to geographic variation, these values differ from farm to farm. The farmers to whom you consult provide you with the values p_c, p_s, p_g, F, f_c, f_s, f_g, W, w_c, w_s and w_g. It is your job to provide them with the ideal number of cattle, sheep and goats such that the amount of profit is maximised and the animals both eat an amount of food less than F and drink an amount of water less than W in a given year. Note that a herd can only have an integer number of a given animal.

Example

A farmer approaches you whose farm has the following parameters

p_c ; p_s ; p_g	7.72 ; 5.78 ; 4.4
F	97.8
f_c ; f_s ; f_g	2.37 ; 7.37 ; 2.61
W	74.4
w_c ; w_s ; w_g	9.24 ; 3.55 ; 7.33

The herd makeup that will maximise the farmer's profit, whilst sticking within her food and water bounds is 3 cattle, 12 sheep and 0 goats.

Question continues on next page

Input and Output

Program input and output will make use of stdio streams (System.in and System.out in Java) i.e. not file I/O.

Input

The first line of the input contains the three numbers p_c , p_s and p_g . The second line contains the number F. The third line contains the numbers f_c , f_s and f_g . The fourth line contains the number W. The fifth line contains the numbers w_c , w_s and w_g . If a line contains multiple numbers, these numbers will be separated by a single space.

Sample Input:

```
7.72 5.78 4.4
97.8
2.37 7.37 2.61
74.4
9.24 3.55 7.33
```

Output

Your output should consist of three integers: the number of cattle, sheep and goats in the optimal herd. The integers should be on the same line and separated by a space. Remember to put a line break at the end of your output. In other words, if you use Java for example, use System.out.println, not System.out.print to print your answer. The automatic marker expects the exactly correct output.

Sample output:

```
3 12 0
```

Constraints

$1 \leq p_c ; p_s ; p_g ; f_c ; f_s ; f_g ; w_c ; w_s ; w_g < 10$
$30 \leq F, W \leq 100$

Scoring

Each test case that is answered correctly will earn 5 points.

File names (Note that case matters)

- Use `herd.c` if you are writing your program in C.
- Use `herd.cpp` if you are writing your program in C++.
- Use `Herd.java` if you are writing your program in Java.
- Use `herd.py` if you are writing your program in Python 3.

Question 2 [50 marks]

File names (Note that case matters)

- Use `roots.c` if you are writing your program in C.
- Use `roots.cpp` if you are writing your program in C++.
- Use `roots.java` if you are writing your program in Java.
- Use `roots.py` if you are writing your program in Python 3.

Problem Description

Modular arithmetic is a special kind of arithmetic on the integers. Instead of the number line extending to infinity in both directions, it wraps around. A very good example of this is the 12-hour time format, where beyond the 12th hour we wrap around to the 1st hour. For example 3 hours from 11 O'clock is 2 O'clock. We can define modular arithmetic more generally with an upper wrap-around limit of n. This would mean that the result of $a + b$ would be rem($a + b$, n), where rem(x, y) is the remainder of x when divided by y. For example, if $n = 10$, then the result of $4 + 8$ would be 2. We can write this as $4 + 8 \equiv 2 \bmod 10$. Multiplication can be defined similarly, where the result of $a * b$ is rem($a * b$, n). For example, if $n = 7$, then the result of $2 * 5$ would be 3 and we could write this as $2 * 5 \equiv 3 \bmod 7$.

Modular arithmetic has important applications in cryptography, including in the ubiquitous RSA algorithm.

Your task is to find all the square roots of a number x under arithmetic modulo n. That is you must find all the integers a, $0 < a < n$, such that $a^2 \equiv x \bmod n$.

Example

$n = 16$ and $x = 4$. There are 4 roots, which are 2, 6, 10 and 14.

Input and Output

Program input and output will make use of stdio streams (System.in and System.out in Java) i.e. not file I/O.

Input

The first line of the input contains the number n and the second line contains the number x.

Sample Input:

```
16
4
```

Output

Your output should consist of space separated integers: all the integers a such that $a^2 \equiv x \bmod n$. The integers should be listed in ascending order.

Sample output:

```
2 6 10 14
```

Constraints

$2 \leq n \leq 100000$

$0 < x < n$

Scoring

Each test case that is answered correctly will earn 5 points.