

Pandamiao

Rar the Cat and Mr. Panda are playing a counting game. Together, they will count integers from **S** to **E** inclusive. For example, if **S** = 7 and **E** = 13, they will count: 7, 8, 9, 10, 11, 12, 13. As this is rather boring, they decide to modify the game to make it more interesting.

Rar the Cat likes the number **R** and Mr. Panda likes the number **P**. The modification to the counting game is as such:

- If the integer is a multiple of **R** but NOT a multiple of **P**, say 'Miao' instead of the integer.
- If the integer is a multiple of **P** but NOT a multiple of **R**, say 'Panda' instead of the integer.
- If the integer is both a multiple of **P** and **R**, say 'Pandamiao' instead of the integer.

However, after deciding on values of **S**, **E**, **R** and **P**, both Rar the Cat and Mr. Panda ran out of time as they must go for an event known as a TA meeting. Can you, a capable CS2040 student, write a program to help them play this modified counting game?

Input

The input contains only 1 line with 4 integers in this order: **S**, **E**, **R** and **P**.

Output

The output should contain **(E-S+1)** lines, printing integers from **S** to **E** in order, one on each line. However, integers that are multiple of **P** only should be replaced with 'Panda', integers that are multiple of **R** only should be replaced with 'Miao' and integers that are multiples of both **P** and **R** should be replaced with 'Pandamiao'.

Limits

- $1 \leq S \leq E \leq 1,000,000,000,000,000,000$ (10^{18})
- $1 \leq R, P \leq 1,000,000,000,000,000,000$ (10^{18})
- However, **S** and **E** will not differ by more than 1000. In other words, $(E-S) \leq 1000$.
- Since the values are very large, you are **recommended to use the 'long' data type**.

Sample Input (pandamiao1.in)	Sample Output (pandamiao1.out)
1 10 2 5	1 Miao 3 Miao Panda Miao 7 Miao 9 Pandamiao

Sample Input (pandamiao2.in)	Sample Output (pandamiao2.out)
98 110 3 2	Panda Miao Panda 101 Pandamiao 103 Panda Miao Panda 107 Pandamiao 109 Panda

Notes:

1. You should develop your program in the subdirectory **ex1** and use the skeleton java file provided. You should not create a new file or rename the file provided.
2. You are free to define your own helper methods and classes (or remove existing ones).
3. Please be reminded that the marking scheme is:
 - a. Public Test Cases (1%) - 1% for passing **all** test cases, 0% otherwise
 - b. Hidden Test Cases (1%) - Partial scoring depending on test cases passed
 - c. Manual Grading (1%)
 - i. Overall Correctness (correctness of algorithm, severity of bugs)
 - ii. Coding Style (meaningful comments, modularity, proper indentation, meaningful method and variable names)
4. Your program will be tested with a time limit of not less than **1 sec** on Codecrunch.

Skeleton File – Pandamiao.java

You are given the below skeleton file `Pandamiao.java`. You should see a non-empty file when you open the skeleton file. Otherwise, you might be in the wrong working directory.

```
/**
 * Name      :
 * Matric. No :
 * PLab Acct. :
 */

import java.util.*;

public class Pandamiao {
    private void run() {
        //implement your "main" method here
    }

    public static void main(String[] args) {
        Pandamiao newPandamiao = new Pandamiao();
        newPandamiao.run();
    }
}
```