

## Problema A

Nombre: Juan Sebastian Vargas  
Codigo: 201215310  
Fecha: 20/05/16

### 1. Algoritmo Solición

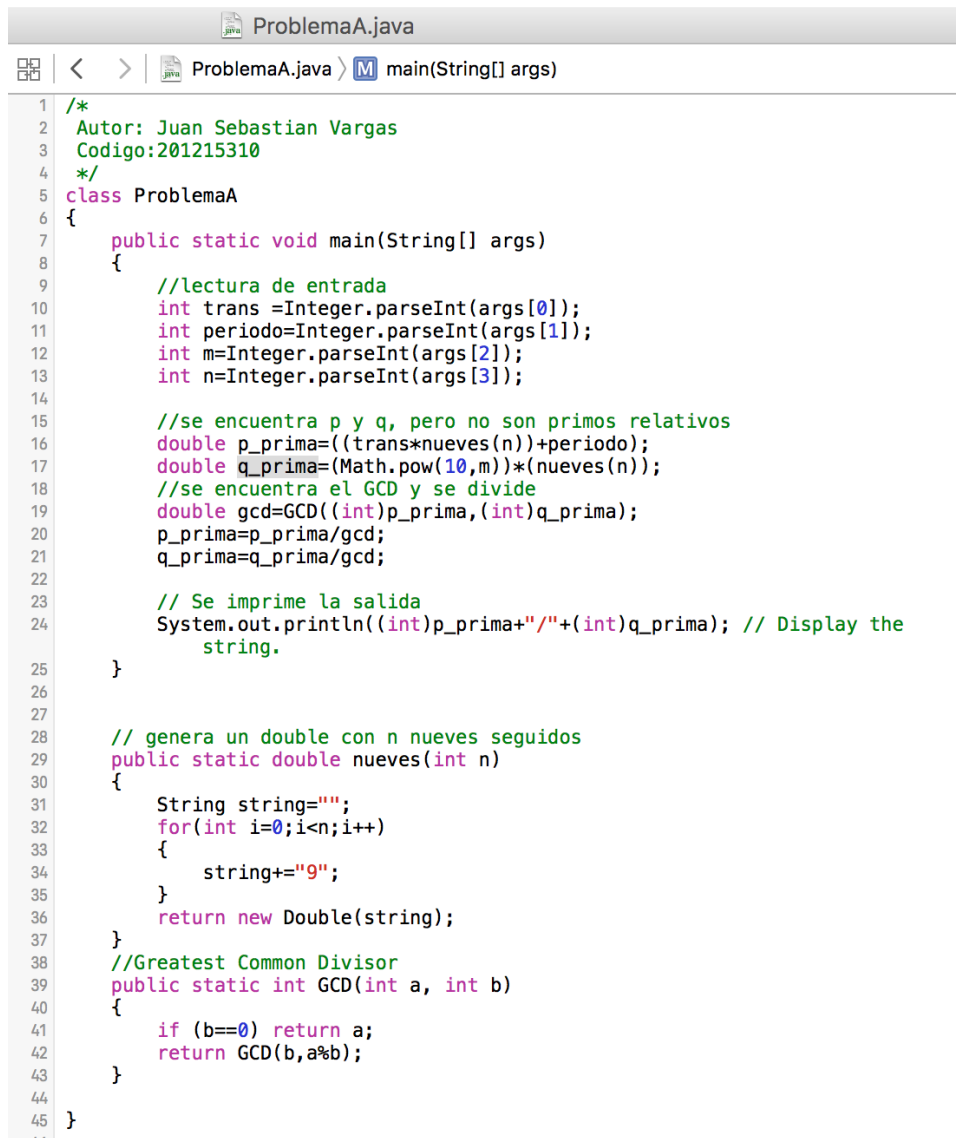
El Algoritmo consta de una lectura de los datos desde  $args[i]$  en el método `main`, y de esta manera se procede a calcular esta ecuación:

$$\frac{p}{q} = \frac{\text{trans}\Xi_n + \text{per}}{10^m \Xi_n} \quad (1)$$

Donde  $\Xi_n$  es una sucesión de  $n$  nuevos seguidos, por ejemplo,  $\Xi_2 = 99$ . Luego se procede a asegurarse que estos dos números sean primos relativos mediante el algoritmo Euclidiano.

### 2. Complejidades

En la lectura de datos no tenemos llamadas repetitivas, ya que solo son necesarias para inicializar el programa. Para resolver la ecuación se requiere de un método auxiliar para generar la cadena de 9's, el cual requiere de  $n$  pasadas para lograr esto, (de existir un método en java mas eficiente el calculo cambiaria). Despues se requiere aplicarles el GCD a los números, el cual tiene  $O(N)$  con  $N = \log(a) + \log(b)$ . Su complejidad espacial es de  $O(1)$  ya que solo se tienen que guardar 7 datos en cualquier caso



```
1  /*
2  Autor: Juan Sebastian Vargas
3 Codigo:201215310
4  */
5  class ProblemaA
6  {
7      public static void main(String[] args)
8      {
9          //lectura de entrada
10         int trans =Integer.parseInt(args[0]);
11         int periodo=Integer.parseInt(args[1]);
12         int m=Integer.parseInt(args[2]);
13         int n=Integer.parseInt(args[3]);
14
15         //se encuentra p y q, pero no son primos relativos
16         double p_prima=((trans*nueves(n))+periodo);
17         double q_prima=(Math.pow(10,m))*(nueves(n));
18         //se encuentra el GCD y se divide
19         double gcd=GCD((int)p_prima,(int)q_prima);
20         p_prima=p_prima/gcd;
21         q_prima=q_prima/gcd;
22
23         // Se imprime la salida
24         System.out.println((int)p_prima+"/"+(int)q_prima); // Display the
           string.
25     }
26
27
28     // genera un double con n nueves seguidos
29     public static double nueves(int n)
30     {
31         String string="";
32         for(int i=0;i<n;i++)
33         {
34             string+="9";
35         }
36         return new Double(string);
37     }
38     //Greatest Common Divisor
39     public static int GCD(int a, int b)
40     {
41         if (b==0) return a;
42         return GCD(b,a%b);
43     }
44
45 }
```

Figura 1: Algoritmo