

División Euclídea

Adriana ha comenzado el primer curso de Ingeniería Informática en la Universidad Complutense de Madrid.

En la asignatura *Matemática Discreta* le han recordado el concepto de División Euclídea de dos números enteros: dados el dividendo a y el divisor b (no nulo), el cociente q y el resto r de la división euclídea se definen como el único par de números que cumplen $a = b * q + r$ siendo $0 \leq r < |b|$.

En la última clase de *Fundamentos de la Programación* ha estudiado cómo escribir expresiones aritméticas, y en particular los operadores $/$ y $\%$ para calcular el cociente y el resto de una división, así que cuando la profesora les pide un programa para calcular el cociente y el resto de una división euclídea, Adriana decide utilizar estos operadores para implementarlo. Con gran sorpresa, se encuentra con que dichos operadores no devuelven lo que ella esperaba para ciertos casos: por ejemplo, cuando $a = -7$ y $b = 3$, en lugar de obtener $q = -3$ y $r = 2$ como esperaba (ya que $-7 = (-3) * 3 + 2$) ¡obtiene $q = -2$ y $r = -1$!

Adriana decide investigar mejor cómo se comportan los operadores con los números negativos y averigua que en C++ sólo cuando el dividendo es positivo devuelven respectivamente el cociente y el resto de la división euclídea. Si el dividendo es negativo el resto es siempre negativo (salvo que sea 0) y cumple $|r| < |b|$. Eso sí, en todos los casos se cumple $a = b * q + r$. Por eso en su ejemplo, como $-7 = (-2) * 3 - 1$, obtiene esos resultados. Aun así, se da cuenta de que con unos pequeños ajustes sigue pudiendo utilizar esos operadores para calcular lo que desea.



Entrada

La entrada comienza con un número indicando la cantidad de casos de prueba que deberán procesarse. Cada caso de prueba está compuesto por dos números enteros: primero el dividendo y luego el divisor.

Salida

Para cada caso de prueba, el programa escribirá en una línea el cociente y el resto de la división euclídea separados por un espacio. En el caso de que el divisor sea 0 se escribirá en la línea: DIV0.

Entrada de ejemplo

```
6
7 3
7 -3
-7 3
-7 -3
7 0
-8 -2
```

Salida de ejemplo

```
2 1
-2 1
-3 2
3 2
DIV0
4 0
```