



EJERCICIOS DE TRAZA

1. Para a=5 y b=3, realiza la traza de las variables indicadas en la tabla:

```
public static void main(String[] args) {  
    int a, b, menor, mcd;  
    Scanner sc = new Scanner(System.in);  
  
    System.out.print("Introduce un número: ");  
    a = sc.nextInt();  
  
    System.out.print("Introduce un número: ");  
    b = sc.nextInt();  
  
    // Calculo el menor  
    menor = (a<b) ? a : b;  
  
    // Hallo el mcd  
    mcd = 1;  
    while (menor>0 && mcd==1){  
        if (a%menor==0 && b%menor==0){  
            mcd = menor;  
        }  
        menor--;  
    }  
  
    System.out.println("El MCD es " + mcd);  
}
```

a	b	menor	mcd	a%menor==0	b%menor==0	a%menor==0 && b%menor==0
5	3	3	1	false	true	false
5	3	2	1	false	false	false
5	3	1	1	true	true	true
5	3	1	1	true	true	true
5	3	0				

2. Para la traza del ejercicio anterior para los valores 4 y 8.

[illegible]



3. A continuación tenemos otra versión del ejercicio mcd. Repite las trazas anteriores.

```
public static void main(String[] args) {  
    int a, b, menor, mcd;  
    Scanner sc = new Scanner(System.in);  
  
    System.out.print("Introduce un número: ");  
    a = sc.nextInt();  
  
    System.out.print("Introduce un número: ");  
    b = sc.nextInt();  
  
    // Calculo el menor  
    mcd = (a<b) ? a : b;  
  
    // Hallo el mcd  
    while (mcd>0 && (a%mcd!=0 || b%mcd!=0)){  
        mcd--;  
    }  
  
    System.out.println("El MCD es " + mcd);  
}
```

a	b	mcd	a%mcd!=0	b%mcd!=0	a%mcd!=0 b%mcd!=0
5	3	3	true	false	true
5	3	2	true	true	true
5	3	1	false	false	false

[illegible]



4. Los dos ejercicios anteriores resuelven el mismo problema. Sin embargo, fíjate que la condición en la que se comprueba que tanto a y b son divididos por un número para ver si es divisor cambia de una a otra:

Versión 1:

```
while (menor>0 && mcd==1){  
    if (a%menor==0 && b%menor==0) {  
        mcd = menor;  
    }  
    menor--;  
}
```

```
a%menor==0 && b  
%menor==0
```

Versión 2:

```
while (mcd>0 && (a%mcd!=0 || b%mcd!=0)) {  
    mcd--;  
}
```

```
a%mcd!=0 || b%mcd!=0
```

- a) Traduce con tus palabras qué se intenta hacer en cada una de las sentencias.

En la primera sentencia podemos apreciar que tenemos un bucle *while* cuyas condiciones son que si menor es mayor que 0 y mcd es igual a 0 pues se ejecute lo que hay dentro del bucle, y dentro de dicho bucle podemos apreciar una condicional *if* que evalúa que si a modulo de menor es igual a 0 y b modulo de menor igual a 0 pues que le asigne a la variable mcd el valor actual de la variable menor.

En la 1, preguntamos si el numerom menor es divisor de a y b.

En la 2 comprobamos que o a o b mp son divisibles por mcd

- b) ¿Por qué funcionan los dos programas si son distintas las condiciones?

Porque en realidad hacen lo mismo

- c) Utilizando el operador not (!), reescribe la condición de la version 2.

```
!(a%mcd==0 && b%mcd==0)
```

- d) ¿Cómo leerías ahora la condición del bucle *while* de la versión 2?

Ejecutaria el bucle mientras mcd sea mayor que 0 y no divida a a y a b.

