



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	0	1	-	-	-



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

a	b	menor	mcd	a%menor==0	b%menor==0	a%menor==0 && b%menor==0
4	8	4	4	true	true	true
4	8	3	4	false	false	false
4	8	2	4	true	true	true
4	8	1	4	true	true	true
4	8	0	4	-	-	-



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

```
public static void main(String[] args) {  
    int a, b, 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 sentencia 1 estamos intentando verificar si al dividir “a” entre “menor” de resto sale 0 y a su vez también si al dividir “b” entre “menor” de resto sale 0.

En la sentencia 2 estamos intentando verificar si al dividir “a” entre “menor” de resto sale 0 o si al dividir “b” entre “menor” de resto sale 0.

- b) ¿Por qué funcionan los dos programas si son distintas las condiciones?
En el primero se está factorizando el menor, y en el segundo ya se da por echo es que el menor ya es el máximo común divisor y lo que comprobamos es si se cumple verdaderamente.

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

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

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

Si no se cumple que el mcd sea igual o menor que 0 o que el resto de la division de a y b entre el mcd sea 0...