

Nombre:	Sergio	Apellidos:	Jiménez Sastre
Asignatura:	Entornos de Desenvolupamiento	Curso y Grupo:	DAW1A
Práctica:	Práctica 4 - Depuración de código en Netbeans		

ÍNDICE

Ejercicio 1.....	2
Ejercicio 2.....	2
Ejercicio 3.....	3
Ejercicio 4.....	4
Ejercicio 4.....	4
Ejercicio 5.....	5
Ejercicio 6.....	5
Ejercicio 7.....	6
Ejercicio 8.....	6
Ejercicio 9.....	7

Nombre:	Sergio	Apellidos:	Jiménez Sastre
Asignatura:	Entornos de Desarrollo	Curso y Grupo:	DAW1A
Práctica:	Práctica 4 - Depuración de código en Netbeans		

Haciendo uso de la herramienta de depuración de Netbeans, encuentra los errores que hay en los siguientes programas. Aprovecha para hacer uso de las diferentes opciones que nos ofrece el depurador (step into, step out, step over, breakpoint, breakpoint condicional, breakpoint con recuento, ejecución hasta el cursor, estado de variables ...).

Al finalizar la práctica debes entregar un documento en el que se indique para cada uno de los programas, el enunciado del programa, las líneas que tenían error y cuál ha sido su modificación de la misma, opciones de depuración de la misma utilizadas , y una o varias capturas de pantalla durante la depuración del código.

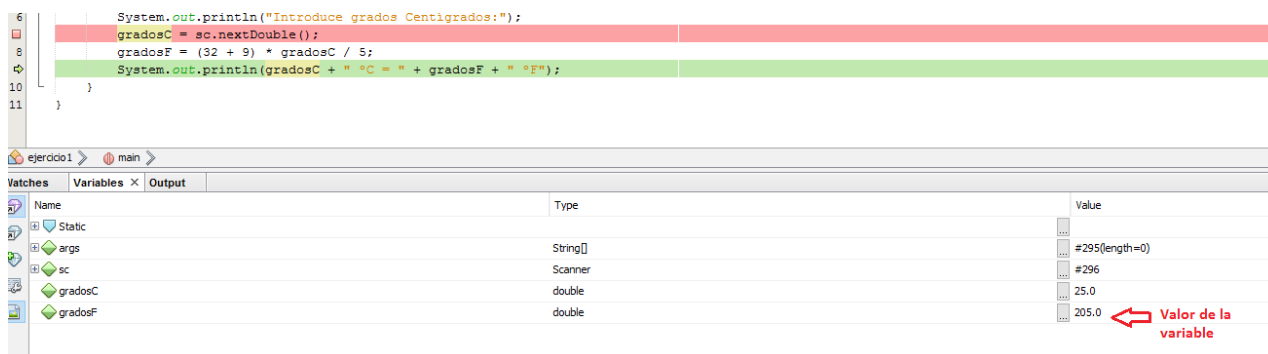
Ejercicio 1

Programa que lea una cantidad de grados centígrados y la pase a grados Fahrenheit. La fórmula correspondiente para pasar de grados centígrados a fahrenheit es: $F = 32 + (9 * C / 5)$. Por ejemplo 25 grados centígrados son 77 grados Fahrenheit.

Línea/s errónea/s: Línea 8.

gradosF = (32 + 9) * gradosC / 5;

Proceso: Hice un breakpoint a partir de donde la variable gradosC va adquiriendo valores (línea 7). Con F8 (Step Over) paso a ejecuta la siguiente línea y observo que el cálculo no es el esperado.



Ejercicio 2

Codifica una función que tenga como parámetros dos números, y que calcule el máximo.

Línea/s errónea/s:

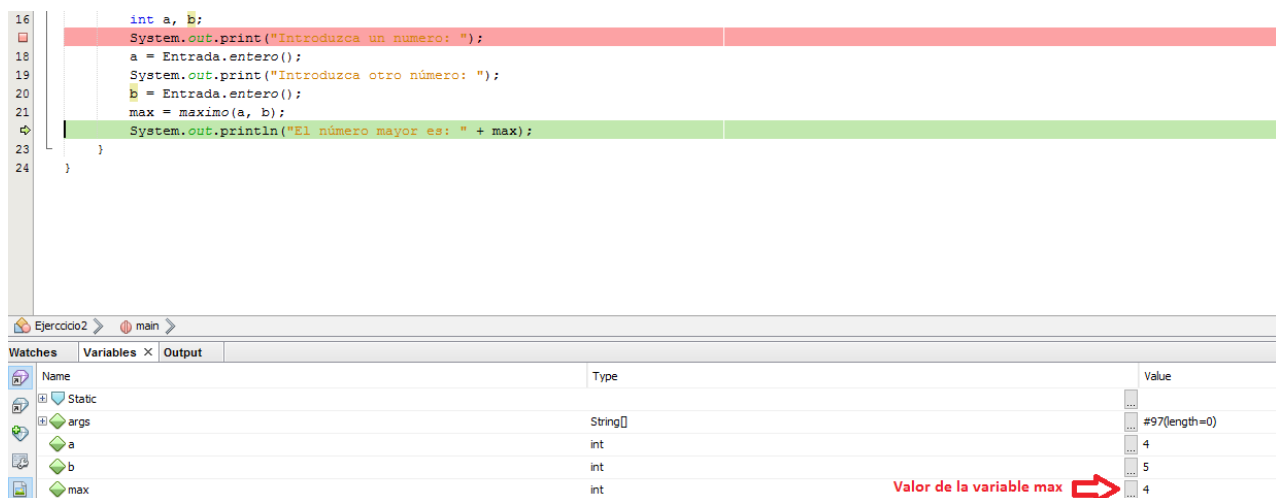
max =maximo (a, b);

Proceso: Hice un breakpoint en el momento de introducir el primer valor y con F8 (Step

Nombre:	Sergio	Apellidos:	Jiménez Sastre
Asignatura:	Entornos de Desarrollo	Curso y Grupo:	DAW1A
Práctica:	Práctica 4 - Depuración de código en Netbeans		

Over) fui viendo qué valores recogían las variables. Al cambiar la línea errónea por el código de abajo el programa hace su funcionalidad correctamente.

```
if (a > b){
    max = a;
}else{
    max = b;
}
```



Ejercicio 3

Una función que calcule el máximo de una tabla de n elementos.

Línea/s errónea/s:

```
for (int i = 2; i < t.length - 2; i++){
```

Proceso: Con un breakpoint en el momento de crear la array de números para ir mirando qué valor obtiene cada variable. Visualizo el valor de i en la primera vuelta y me doy cuenta de que mete un número aleatorio empezando por esa posición. La solución fue inicializar en 0 y no en 2.

```
for (int i = 2; i < t.length - 2; i++){
```

Nombre: Sergio **Apellidos:** Jiménez Sastre
Asignatura: Entornos de Desarrollo **Curso y Grupo:** DAW1A
Práctica: Práctica 4 - Depuración de código en Netbeans

```
public static void main(String[] args) {
    int max;
    int t[];

    t = new int[6];
    for (int i = 2; i < t.length - 2; i++) { // llenamos :
        t[i] = (int) (Math.random() * 100 + 1);
        System.out.println("Los valores son:");
    }
    for (int i = 0; i < t.length; i++) {
        System.out.print(t[i] + " ");
    }
    max = maximo(t);
    System.out.println("\nEl número mayor es: " + max);
}
```

pactica1.Ejercicio3 > main > for (int i = 2; i < t.length - 2; i++) >			
Watches	Variables	Output	
Name	Type	Value	
Static			
args	String[]	#97(length=0)	
t	int[]	#99(length=6)	
i	int	2	Valor variable i

Ejercicio 4

Función a la que se le pasan dos enteros y muestra todos los números comprendidos entre ellos, inclusive.

Línea/s errónea/s:

for (int i = menor + 1; i < mayor; i++) {

```
public class Ejercicio4 {
    static void mostrar(int a, int b) {
        int mayor, menor;
        // desconocemos el orden en el que vienen a y b.
        // Lo que haremos es poner los valores correctos en mayor, menor.

        if (a > b) { // a es el mayor. Se podría utilizar la función maximo()
            mayor = b;
            menor = a;
        } else { // en este caso b será el mayor mayor=b;
            menor = b;
        }
        for (int i = menor + 1; i < mayor; i++) {
            System.out.print(i + " ");
        }
        System.out.println();
    }

    public static void main(String[] args) {
        int a, b;
        System.out.print("Introduzca primer numero: ");
        a = Entrada.entero();
        System.out.print("Introduzca segundo numero: ");
        b = Entrada.entero();

        mostrar(a, b);
    }
}
```

```
public class Ejercicio4 {
    static void mostrar(int a, int b) {
        int mayor, menor;
        // desconocemos el orden en el que vienen a y b.
        // Lo que haremos es poner los valores correctos en mayor, menor.

        if (a < b) { // a es el mayor. Se podría utilizar la función maximo()
            mayor = b;
            menor = a;
        } else { // en este caso b será el mayor mayor=b;
            menor = b;
            mayor = a;
        }
        for (int i = menor + 1; i < mayor; i++) {
            System.out.print(i + " ");
        }
        System.out.println();
    }

    public static void main(String[] args) {
        int a, b;
        System.out.print("Introduzca primer numero: ");
        a = Entrada.entero();
        System.out.print("Introduzca segundo numero: ");
        b = Entrada.entero();

        mostrar(a, b);
    }
}
```

Proceso: En primer lugar hay que asignar valor a la variable mayor en el else. Y en segundo lugar cambiar el signo > a < en la condición de la forma siguiente →

Nombre:	Sergio	Apellidos:	Jiménez Sastre
Asignatura:	Entornos de Desarrollo	Curso y Grupo:	DAW1A
Práctica:	Práctica 4 - Depuración de código en Netbeans		

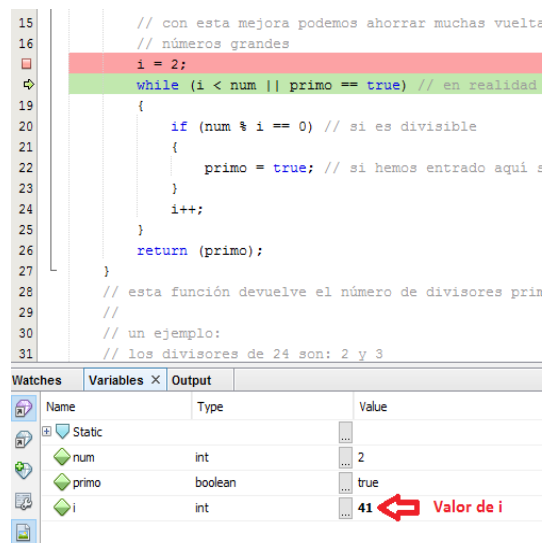
Ejercicio 5

Módulo al que se le pasa un número entero y devuelve el número de divisores primos que tiene.

Proceso: Hice un breakpoint y utilizando F8 (Step over) y Ctrl+F7 (Step out) me fijé en que las vueltas que da el bucle while son infinitas (el valor de i no paraba de aumentar).

Línea/s errónea/s:

```
if (num % i == 0) {
    primo = true;
}
i++;
```



```

15 // con esta mejora podemos ahorrar muchas vueltas
16 // números grandes
17 i = 2;
18 while (i < num || primo == true) // en realidad
19 {
20     if (num % i == 0) // si es divisible
21     {
22         primo = true; // si hemos entrado aquí :
23     }
24     i++;
25 }
26 return (primo);
27 }
28 // esta función devuelve el número de divisores primos
29 //
30 // un ejemplo:
31 // los divisores de 24 son: 2 y 3

```

Name	Type	Value
Static		
num	int	2
primo	boolean	true
i	int	41

Solución: En la condición if que se encuentra dentro del bucle while cambié la línea **primo = true;** a **primo = false;**, de esta forma el bucle ya no es infinito.

Ejercicio 6

Realizar una función que devuelve una tabla con los divisores.

Línea/s errónea/s:

```
if (es_primo(i) && num % i == 0)
```

Nombre: Sergio **Apellidos:** Jiménez Sastre
Asignatura: Entornos de Desarrollo **Curso y Grupo:** DAW1A
Práctica: Práctica 4 - Depuración de código en Netbeans

Solución: En la condición if que se encuentra dentro del bucle for eliminé la llamada al método *es_primo()* ya que en este caso no nos interesa hacer esta comprobación.

```

20 num_div = num_divisores(num);
21 div = new int[num_div];
22 for (int i = 1; i <= num; i++) {
23     if (num % i == 0) // si i es
24     {
25         div[cont] = i; // increme
26         cont++;
27     }
28 }
29 return (div);
30 }

```

Ejercicio 7

Escribir una función que calcule el máximo común divisor de dos números.

Línea/s errónea/s:

if (a % i == 0 || b % i == 0)

Proceso y solución: En la condición if que se encuentra dentro del bucle for la operación lógica or (||) causaba que no nos mostrara el máximo común divisor. Para que dicho número sea el máximo común divisor hay que cambiarlo por la operación lógica and (&&), de esta manera ↓

```

for (int i = 2; i < min; i++) {
    if (a % i == 0 && b % i == 0)
    {
        mcd = i;
    }
}
return (mcd);

```

```

11 static int max_comun_divisor(int a, int b) {
12     int mcd = 1;
13     int min;
14     min = minimo(a, b);
15     mcd = 1; // existe un mcd seguro, el 1, q
16     for (int i = 2; i < min; i++) {
17         if (a % i == 0 && b % i == 0) // si a
18         {
19             mcd = i; // i será el nuevo mcd.
20         }
21     }
22     return (mcd);
23 }
24
25 static int minimo(int a, int b) {
26     int min;
27     if (a > b) {
28         min = b;
29     } else {
30         min = a;
31     }
32 }

```

Name	Type	Value
Static		
a	int	12
b	int	18
mcd	int	1
min	int	12
i	int	2

Ejercicio 8

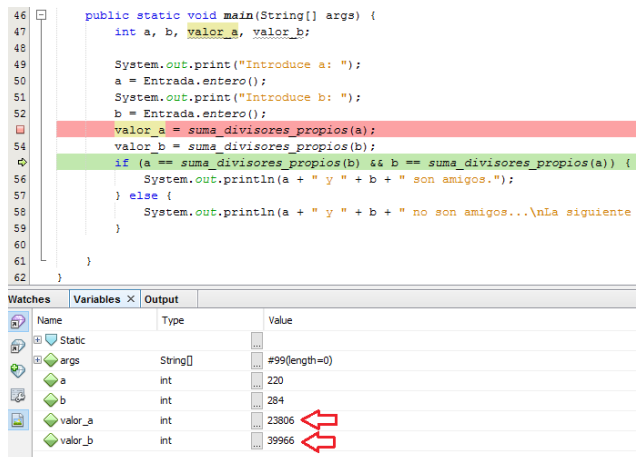
Escriba una función que decida si dos números enteros positivos son amigos. Dos números son amigos, si la suma de sus divisores (distintos de ellos mismos) son iguales.

Nombre:	Sergio	Apellidos:	Jiménez Sastre
Asignatura:	Entornos de Desarrollo	Curso y Grupo:	DAW1A
Práctica:	Práctica 4 - Depuración de código en Netbeans		

Línea/s errónea/s:

for (int i = 3; i < num; i++)
if (num % i != 0)

Proceso: En primer lugar cree dos variables para verificar qué devuelven `suma_divisores_propios(a)` y `suma_divisores_propios(b)`, para eso hice un breakpoint y me fijé en el valor dado.



```
static int suma_divisores_propios(int num) {
    int suma;
    suma = 0;
    for (int i = 1; i < num; i++)
    {
        if (num % i == 0)
        {
            suma += i;
        }
    }
    return (suma);
}
```

Solución: Dentro del método que nos devuelve la suma de los divisores volví a hacer un breakpoint y me percaté de un par de errores en un for: en primer lugar inicialicé `i` en 1, en segundo lugar en la condición `if` que se encuentra dentro del bucle `for` está la operación lógica diferente (`!=`) y la cambié por la operación lógica igual (`==`).

Ejercicio 9

Escriba una función que sume los n primeros números impares.

Línea/s errónea/s:

suma += 2 * i - 2;

Proceso: Hice un breakpoint justo antes de que la variable `suma` obtuviera algún valor, al darme cuenta que obtenía mal el valor le añadí un condicional para averiguar si el número es impar o no.

```
if (i % 2 != 0){
    suma += i;
}
```

Nombre: Sergio **Apellidos:** Jiménez Sastre
Asignatura: Entornos de Desarrollo **Curso y Grupo:** DAW1A
Práctica: Práctica 4 - Depuración de código en Netbeans

```
public class Ejercicio9 {  
    static int suma_n_impares(int n) {  
        int suma = 0;  
        for (int i = 1; i < n; i++) {  
            suma += 2 * i - 2; // así calculamos el i-ésimo impar  
        }  
        return (suma);  
    }  
  
    public static void main(String[] args) {  
        int n;  
        System.out.print("Introduzca un número: ");  
        n = Entrada.entero();  
        System.out.println("La suma de los " + n + " primeros impares es: " + suma_n_impares(n));  
    }  
}
```

pactica1.Ejercicio9 suma_n_impares			
Watches Variables × Output			
	Name	Type	Value
	Static		...
	n	int	2
	suma	int	0