

El objetivo de esta práctica es detectar posibles errores de ejecución de un programa que compila correctamente, pudiendo obtener como resultado salidas incorrectas o la detención del programa.

Existen dos técnicas para la solución de estos errores, como son: utilización de trazas y utilización del depurador.

1. Hacer una traza “a mano” sobre la tabla siguiente de la posible ejecución de este programa:

```
#include <stdio.h>
int main ()
{
    int n=1, fin=0;
    while (n<=4 || !fin )
    {
        printf("Buenos ");
        n++;
        if (n==2)
            fin=1;
    }
    printf("Dias");
    return 0;
}
```

TRAZA:

Línea de código	n	fin	n <= 4 !fin	Resultado en la pantalla
1	1	0		
2			1	
3				Buenos
4	2			
5				
6		1		
7			1	
4				Buenos
1	1	0		
2			1	
3				Buenos
4	3			
5				
6		0		
7			1	
4				Buenos
7				

2. Editar el programa, compilarlo y depurarlo ejecutando sentencia a sentencia visualizando al mismo tiempo el valor de las variables en la ventana Watches del Debug (ver el documento de [introducción al depurador CodeBlocks](#)):

Capturar la pantalla donde se vea la ventana Watches y pegarla a continuación.

	Function arguments		
☐	Locals		
	n	4194432	
	fin	2678784	
	0	0	int
	k	nt context!	...

3. Dado el siguiente código en lenguaje C, añadir las sentencias de escritura necesarias de forma que se complete en la consola la traza de la ejecución, mostrando los valores que contienen las variables en cada momento de la ejecución. Incluya en el recuadro una captura de pantalla de la ejecución del programa.

```
#include <stdio.h>
int main()
{
    int a,b;
    printf("\nTraza 1: Valor de las variables 'a' y 'b' antes de inicializarse: %d %d\n", a, b);
    a = 0;
    b = 0;
    printf("Traza 2: Valor de las variables 'a' y 'b' tras inicializarse: %d %d\n", a,b);
    scanf("%d",&a);
    scanf("%d",&b);
    printf("Traza 3: Valor de las variables 'a' y 'b' tras leer un valor por consola: %d %d\n", a, b);
    // intercambiar los valores de las variables a y b
    //...
    // asignar a la variable b el doble del valor de la variable a
    //...
    return 0;
}
```

```
Traza 1: Valor de las variables 'a' y 'b' antes de inicializarse: 4194432 4100096
Traza 2: Valor de las variables 'a' y 'b' tras inicializarse: 1 0
a
Traza 3: Valor de las variables 'a' y 'b' tras leer un valor por consola: 1 0
Intercambio de variables: 01
La variable b es el doble del valor e 'a', la variable 'b' es: 2

Process returned 0 (0x0)   execution time : 2.066 s
Press any key to continue.
```

4. Utilizando el código en lenguaje C anterior, utilizar el depurador para realizar todas las trazas señaladas anteriormente sin necesidad de mostrar los mensajes por consola (usando la ventana Watches). Captura la pantalla donde se vea la ventana Watches y pegarla a continuación.

Watches			
Function arguments			
Locals			
n	4194432		
fin	2678784		
0	0		int
k	nt context!	...	

5. Escribe un programa que reciba por teclado un número entero comprendido entre 0 y 10, calcule su factorial y lo muestre por pantalla. El programa seguirá solicitando números hasta que se introduzca un -1 para terminar.

Copia a continuación el programa desarrollado.

```
#include <stdio.h>
int main()
{
    int i,num,fact=1,cont_salir;
    do{
        printf("Introduce un numero para calcular el facotiral: ");
        scanf("%d", &num);
        fflush(stdin);
        for(i=1;i<=num;i++){
            fact=fact * i;
        }
        printf("Su numero es: %d   Desea continuar ? 1/SI---0/NO: ",fact);
        scanf("%d", &cont_salir);
        i=0; ;fact=1;
    }while(cont_salir!=-1);
    return 0;
}
```

Verifica su correcto funcionamiento utilizando el depurador. Indica qué variables consideras necesario comprobar.

No he podido ejecutar watches en ninguno de los ejercicios exceptuando los primeros ya que los hice en clase, tengo un error con el codeblock pero no sé como solucionarlo.

6. Escribe un programa que reciba por teclado dos números enteros comprendidos entre 1 y 1000, calcule su mínimo común múltiplo y lo muestre por pantalla. Una vez escrito el resultado, el programa preguntará si se desea seguir introduciendo números, dando como opciones:

- 0 (finalizar la ejecución)
- 1 (continuar introduciendo datos).

Copia a continuación el programa desarrollado.

```
#include<stdio.h>

int main(){

int n1=0,mcd=0,n2=0;
    int mcm=0;
    int aux=0;
    int opcion;
    do {
        do {
            printf("Introduce primer numero : ");
            scanf("%d",&n1);
            printf("Introduce segundo numero : ");
            scanf("%d",&n2);
        }while(n1<1 || n1>1000 || n2<1 || n2>1000);

        if(n1<n2) {
            aux=n1;
        }else {
            aux=n2;
        }

        for (int i=1;i<=aux;i++) {

            if(n1%i==0 && n2%i==0) {

                mcd=i;
                mcm=(n1*n2)/mcd;

            }

        }

        printf("el mcm es : %d ",mcm);

        do {
            printf("Introduce opcion: ");
            printf("0.Finalizar Programa; ");
            printf("1.Continuar introduciendo datos ");
            scanf("%d",&opcion);
        }while(opcion>1 || opcion<0);

    }while(opcion!=0);
    return 0;
```

```
}
```

Verifica su correcto funcionamiento utilizando el depurador. Al menos hay que comprobar el resultado en los siguientes casos:

- 7 y 13
- 15 y 60
- -7 y 44