

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 n1, n2, n3;
    n1=10;
    n2=20;
    n3=30;
    n1= n1+n2+n3;
    n2=n3 / 10 + 4 * n1;
    n3++;
    n1=n2=n3+1;
    return 0;
}
```

TRAZA:

Línea de código	n1	n2	n3
4	indeterminado	indeterminado	indeterminado
5	10	indeterminado	indeterminado
6	10	20	indeterminado
7	10	20	30
8	60	20	30
9	60	243	30
10	60	243	31
11	32	32	31

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.

The program being debugged is as follows:

```

1  #include <stdio.h>
2  int main ()
3  {
4      int n1, n2, n3;
5      n1=10;
6      n2=20;
7      n3=30;
8      n1= n1+n2+n3;
9      n2=n3 / 10 + 4 * n1;
10     n3++;
11     n1=n2=n3+1;
12     return 0;
13 }
14

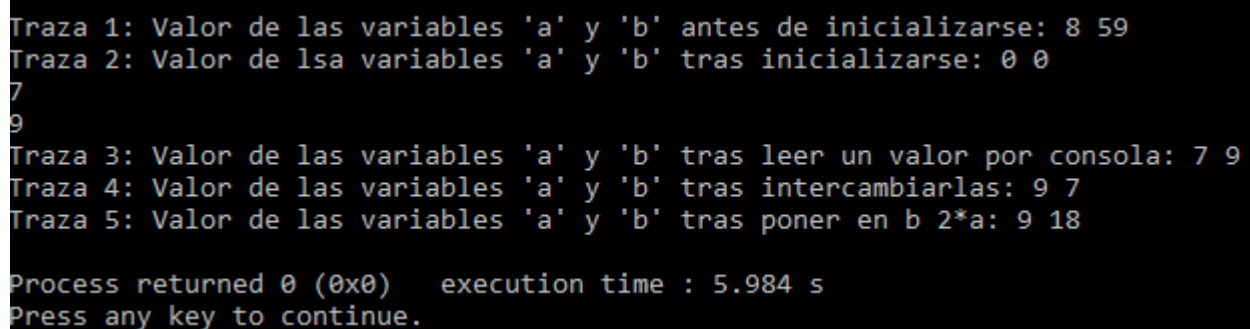
```

The following table summarizes the state of the variables `n1`, `n2`, and `n3` as shown in the Watch window across the 10 screenshots:

Screenshot	Line of Code Executed	n1	n2	n3
1	Line 4: <code>int n1, n2, n3;</code>	4199415	6422368	0
2	Line 5: <code>n1=10;</code>	10	20	0
3	Line 6: <code>n2=20;</code>	10	20	0
4	Line 7: <code>n3=30;</code>	10	20	30
5	Line 8: <code>n1= n1+n2+n3;</code>	60	20	30
6	Line 9: <code>n2=n3 / 10 + 4 * n1;</code>	60	243	30
7	Line 10: <code>n3++;</code>	60	243	31
8	Line 11: <code>n1=n2=n3+1;</code>	32	32	31
9	Line 12: <code>return 0;</code>	32	32	31
10	End of program	32	32	31

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 la variable 'a' y 'b' tras inicializarse: %d %d\n",
a,b);
    scanf("%d",&a);
    scanf("%d",&b);
    printf("Traza 3: Valor de la variable '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: 8 59
Traza 2: Valor de lsa variables 'a' y 'b' tras inicializarse: 0 0
7
9
Traza 3: Valor de las variables 'a' y 'b' tras leer un valor por consola: 7 9
Traza 4: Valor de las variables 'a' y 'b' tras intercambiarlas: 9 7
Traza 5: Valor de las variables 'a' y 'b' tras poner en b 2*a: 9 18

Process returned 0 (0x0)   execution time : 5.984 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). Capturar la pantalla donde se vea la ventana Watches y pegarla a continuación.

The screenshots show the 'Watches' window in a debugger, displaying the values of local variables `a`, `b`, and `c` at different points in the execution of the following C program:

```
#include <stdio.h>
int main()
{
    int a,b, c;
    // printf("\nTraza 1: Valo
    a = 0;
    b = 0;
    // printf("Traza 2: Valo
    scanf("%d",&a);
    scanf("%d",&b);
    // printf("Traza 3: Valo
    c = a;
    a = b;
    b = c;
    // printf("Traza 4: Valo
    b = 2*a;
    // printf("Traza 5: Valo
    return 0;
}
```

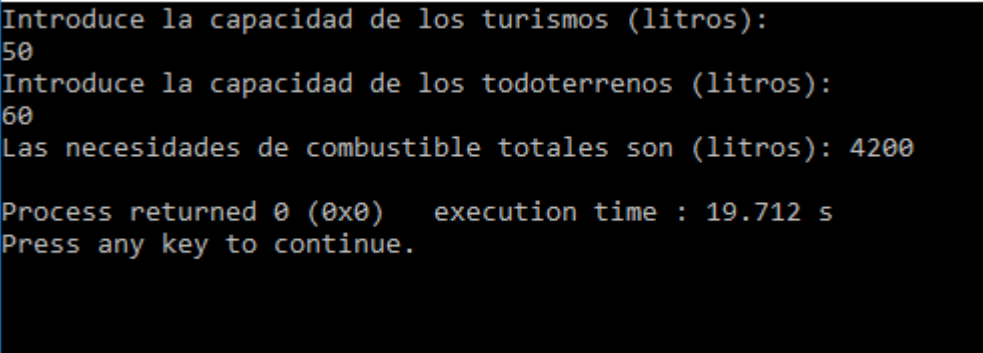
The five screenshots show the state of the variables at different points in the execution:

- Screenshot 1:** Initial state. `a = 6422368`, `b = 0`, `c = 4199415`.
- Screenshot 2:** After the first `scanf` call. `a = 0`, `b = 0`, `c = 4199415`.
- Screenshot 3:** After the second `scanf` call. `a = 7`, `b = 9`, `c = 4199415`.
- Screenshot 4:** After the first assignment `c = a`. `a = 9`, `b = 7`, `c = 7`.
- Screenshot 5:** After the second assignment `b = c`. `a = 9`, `b = 18`, `c = 7`.

5. Una empresa que trabaja con vehículos desea calcular las necesidades de combustible (cantidad de combustible necesario para llenar los depósitos de todos sus vehículos) para lo cual nos han facilitado que dispone de 30 turismos y 45 todoterrenos, sabiendo que la capacidad del depósito es diferente. Se desea crear un programa en C para que puedan realizar el cálculo de forma automatizada.

Inserte a continuación el código desarrollado y una captura de pantalla de la ejecución.

```
#include <stdio.h>
#include <stdlib.h>
int main()
{
    // Numero de vehículos
    int turismos = 30, todoterrenos = 45;
    // Capacidades de los vehículos
    int capTur, capTodot;
    printf ("Introduce la capacidad de los turismos (litros): \n");
    scanf ("%d", &capTur);
    printf ("Introduce la capacidad de los todoterrenos (litros): \n");
    scanf ("%d", &capTodot);
    //Cálculo de necesidades de combustible
    printf ("Las necesidades de combustible totales son (litros): %d  \n", turismos *
capTur + todoterrenos * capTodot);
    return 0;
}
```



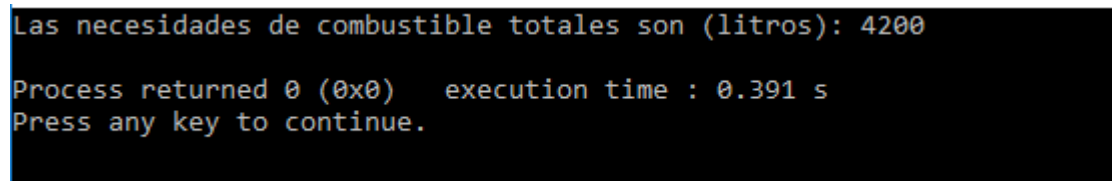
```
Introduce la capacidad de los turismos (litros):
50
Introduce la capacidad de los todoterrenos (litros):
60
Las necesidades de combustible totales son (litros): 4200

Process returned 0 (0x0)   execution time : 19.712 s
Press any key to continue.
```

6. Se desea realizar el mismo programa en C que el apartado anterior pero con el siguiente cambio: los datos de la capacidad de los depósitos tanto de los turismos como de los todoterrenos se encuentra en un fichero denominado "datos.txt".

Inserte a continuación el código desarrollado y una captura de pantalla de la ejecución.

```
#include <stdio.h>
#include <stdlib.h>
int main()
{
    // Numero de vehículos
    int turismos = 30, todoterrenos = 45;
    // Capacidades de los vehículos
    int capTur, capTodot;
    FILE *fichero;
    // se abre el fichero para leer los datos que contiene
    fichero = fopen("datos.txt", "r");
    if(fichero == NULL)
        printf("no se puede abrir el fichero");
    else
    {
        // se leen del fichero las capacidades de los turismos y todoterrenos
        fscanf(fichero,"%d%d", &capTur, &capTodot);
        //Cálculo de necesidades de combustible
        printf ("Las necesidades de combustible totales son (litros): %d \n", turismos *
capTur + todoterrenos * capTodot);
        close(fichero);
    }
    return 0;
}
```



```
Las necesidades de combustible totales son (litros): 4200
Process returned 0 (0x0)   execution time : 0.391 s
Press any key to continue.
```