
Fundamentos de Programación

Sesión 9

Actividades a realizar en casa

Actividad: Resolución de problemas.

Resolved los siguientes problemas de la relación 4. Recordad que, **ANTES** del inicio de esta sesión en el aula de ordenadores, hay que subir las soluciones a PRADO. Se debe subir un fichero llamado sesion9.zip que incluya todos los ficheros cpp (y solamente estos).

- 3 (Confirmación)
- 4 (Minúscula)
- 6 (MCM)
- 7 (Fecha válida)
- 11 (Números amigos)

Actividades a realizar en las aulas de ordenadores

Depuración de funciones

En la sesión 4 trabajamos sobre la depuración de programas usando Dev C++. Entonces no conocíamos cómo escribir funciones y no pudimos sacar partido a todas las opciones de depuración. En esta sesión de prácticas vamos a trabajar sobre la manera en la que se puede monitorizar la ejecución de un programa que incluye funciones.

Usaremos como ejemplo la función Combinatorio. En primer lugar, crearemos en U:\FP la carpeta 9_Combinatorio y copiaremos dentro el archivo 9_FuncionesCombinatorio.cpp disponible en PRADO.

Antes de empezar con las tareas de depuración observaremos el explorador de clases. Para acceder a él, basta con seleccionar Ver | Ir al Explorador de Clases. En la Figura 10 puedes observar que el explorador de clases muestra, para este programa, información acerca de las funciones contenidas en el fichero fuente abierto en el editor.

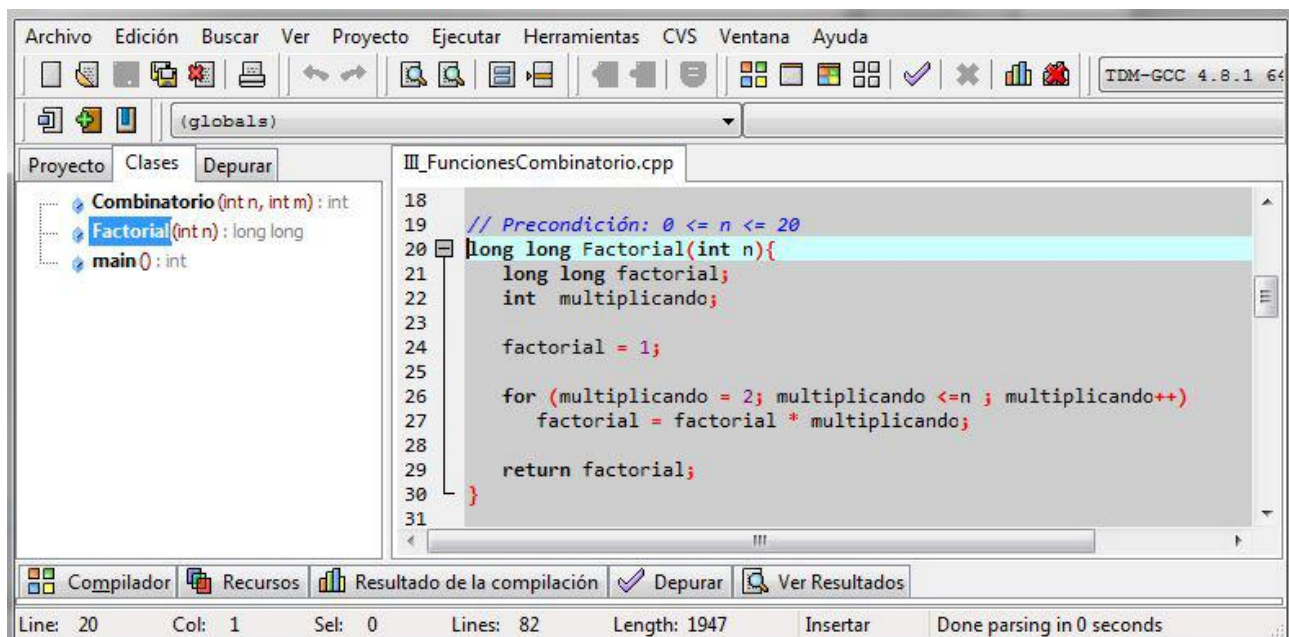


Figura 10: Explorador de clases

Para cada función muestra su nombre, los parámetros formales y su tipo, así como el tipo de la función (el tipo del valor devuelto). Se muestran todas las funciones en orden alfabético. Haciendo click sobre el nombre de cualquier función en el explorador, en el editor vemos el código de la función seleccionada. Éste es una manera rápida de acceder al código de cualquier función en nuestros programas.

El proceso de depuración se inicia de la manera habitual:

1. Fijar un punto de ruptura.
2. Comenzar la depuración.

Fijaremos un punto de ruptura, por ejemplo, en la línea del main

```
combinatorio = Combinatorio(total_a_elegir, elegidos);
```

y comenzamos la depuración.

El programa se ejecuta hasta llegar dicha línea, donde se detiene. Ahora podemos monitorizar su ejecución usando los botones disponibles en la zona inferior, bajo la pestaña Depurar.

Parar ejecución: Detener la depuración (y ejecución) del programa.

Siguiente Paso. F7: Ejecuta la siguiente instrucción. Si se trata de una llamada a una función, la ejecuta y continúa con la siguiente instrucción, sin entrar a ejecutar las instrucciones internas de la función.

Avanzar Paso a paso. F8: Ejecuta la siguiente instrucción. Si se trata de una llamada a una función, entra en la función y ejecuta la primera instrucción de la función, continuando la depuración dentro de la función.

Saltar Paso.: Ejecuta todas las instrucciones hasta encontrar un nuevo punto de ruptura, o llegar al final del programa.

En la línea en la que está situado el punto de ruptura, si se pulsara **Siguiente Paso** se ejecuta completamente esa línea: la llamada a la función `Combinatorio` y la instrucción de asignación, pasando el control a la línea siguiente del main. Observad cómo la variable `combinatorio` se ha actualizado correctamente.

Durante la ejecución de una función pueden añadirse a la lista de variables monitorizadas cualquiera de las variables locales de la función (incluidas los parámetros formales, por supuesto). Al finalizar la ejecución de la función y dejar de estar activas las variables locales de la función veremos un mensaje de error en estas variables.

La ejecución completa de línea siguiente conlleva la ejecución de dos llamadas a la función `Factorial`:

```
denominador = Factorial(m) * Factorial(n - m);
```

En este punto,

- si se pulsa **Siguiente Paso** se completa la ejecución de esa línea y se cede el control a la línea siguiente. Observad que la variable local `combinatorio` contiene el valor ya calculado.
- si se pulsa **Avanzar Paso a paso** se entra a ejecutar la función `Factorial`, pasando el control a la primera instrucción de esa función

Continúa monitorizando la ejecución del programa como quieras. No olvides probar a establecer un punto de ruptura dentro de una función y observar qué ocurre cuando se pulsa el botón **Siguiente Paso** ¿se detiene la ejecución en el punto de ruptura o lo ignora?

Actividad: Resolución de problemas.

Resolved los siguientes problemas de la relación 4.

12 (Calculadora)

13 (Inverso)

14 (Perfectos)