當編譯包含前面定義的程式時（您可以方便地將其放在標頭檔，並載入到程式），您可以定義 DEBON 或不定義。如果您編譯 prog.c 如下：

```
$ gcc prog.c
```

它基於前面前置處理器指令中顯示的 #else 子句，在 DEBUG 的空定義中進行編譯。另一方面，如果您編譯程式如下：

```
$ gcc -D DEBON prog.c
```

基於除錯級別呼叫 fprintf 的 DEBUG 巨集，並與其餘程式碼一起被編譯。在執行時，如果已在除錯程式碼中編譯，可以選擇除錯級別。如上所述，這可以使用命令列選項來完成，如下所示：

```
$ a.out -d3
```

此處的除錯級別設置為 3。假定於程式中處理此命令列參數，並將除錯級別儲存在名為 Debug 的變數（或許是全域變數）。在這種情況下，只有指定級別為 3 或更大的 DEBUG 巨集才會執行 fprintf 呼叫。

請注意，a.out -d0 將除錯級別設置為 0，即使得除錯程式碼仍然存在，但也不會產生除錯輸出。總而言之，您在這裡看到了一個雙層（two-tiered）除錯方案：除錯程式碼可以在程式碼中或程式碼之外編譯，並且在編譯時可以設置不同的除錯級別，以產生不同數量的除錯輸出。

使用 gdb 除錯程式

gdb 是一個強大的交談式除錯器（interactive debugger），經常用於除錯以 GNU 的 gcc 編譯器所編譯的程式。它允許您執行程式，在預定位置停止，顯示和/或設定變數，以及繼續執行。它允許您追蹤程式的執行，甚至一次只執行一行。gdb 還具有決定核心轉儲（core dump）位置的機制。核心轉儲是因為某些異常事件而發生，如除以零或嘗試存取超過陣列的大小。這將導致建立一個名為 core 的檔案，該檔案包含程式終止時執行過程記憶體的內容之快照。[1]

你撰寫的 C 程式必須使用 gcc 編譯器的-g 選項加以編譯，才能利用 gdb 的功能。-g 選項會使 C 編譯器添加額外訊息到輸出檔案，包括變數、結構型態、原始檔名，以及 C 敘述所對映的機器碼。

[1] 您的系統可能被配置為不可使用自動建立此core檔案，通常是因為這些檔案的大小。有時，這與所建立檔案的大小有關，可以使用ulimit指令更改之。

範例程式 17.4 顯示了一個試圖存取超過陣列大小的程式。

範例程式 17.4 一個使用 gdb 的簡單程式

```
#include <stdio.h>

int main (void)
{
     const int  data[5] = {1, 2, 3, 4, 5};
     int  i, sum;

     for (i = 0; i >= 0; ++i)
         sum += data[i];

     printf ("sum = %i\n", sum);

     return 0;
}
```

下面是當從終端器在 Mac OS X 系統執行程式時發生的情況（在其它系統上，執行程式時可能會顯示不同的訊息）：

```
$ a.out
```

使用 gdb 嘗試追蹤錯誤。這是一個經過設計的例子，只是用來說明而已。

首先，請確保使用 -g 選項編譯程式。接著，在可執行檔上啟動 gdb，預設的情況下它是 a.out。這可能會導致系統顯示一些開場白的訊息：

```
$ gcc -g p18.4.c          重新編譯使其含有除錯訊息給 gdb
$ gdb a.out               在可執行檔上啟動 gdb
GNU gdb 5.3-20030128 (Apple version gdb-309) (Thu Dec  4 15:41:30 GMT 2003)
Copyright 2003 Free Software Foundation, Inc.
GDB is free software, covered by the GNU General Public License, and you are
welcome to change it and/or distribute copies of it under certain conditions.
Type "show copying" to see the conditions.
There is absolutely no warranty for GDB. Type "show warranty" for details.
This GDB was configured as "powerpc-apple-darwin".
Reading symbols for shared libraries .. done
```

當 gdb 準備好接收指令時，它顯示一個（gdb）提示符號。在我們的簡單範例中，只需要輸入 run 指令來告訴它執行程式。這會使得 gdb 開始執行程式，直到執行完畢或發生異常事件：

```
(gdb) run
Starting program: /Users/stevekochan/MySrc/c/a.out
Reading symbols for shared libraries . done

Program received signal EXC_BAD_ACCESS, Could not access memory.
```