**Step 1: Compile with Debug Information**

First, ensure your assembly code is compiled with debug information. This makes it easier to step through the code in the debugger. For assembly programs, you'll usually pass the **-g** flag to **as** and **ld**:

**as -g -o program.o program.s**

**ld -o program program.o**

**Step 2: Start GDB**

Launch GDB with your compiled program:

**gdb ./program**

**Step 3: Set Breakpoints**

**break MultiplyArrayElements**

**Step 4: Run the Program**

**run**

**Step 5: Step Through the Code**

Once you hit a breakpoint, you can step through your code using **stepi** (step instruction) to move one assembly instruction at a time, or **nexti** to step over function calls. Use **info registers** to inspect the current values of registers:

**stepi**

**info registers**

**Step 6: Inspect Memory**

To inspect the contents of memory, use the x command followed by an address. For example, to inspect the contents of the memory pointed to by x1, you can do:

**x/gx $x1**

The **/gx** specifies that GDB should display the memory content as a hexadecimal and treat it as a giant (64-bit) value. Adjust the format according to what you're inspecting (**/x** for hexadecimal, **/d** for decimal, **/s** for a string, and the size **g** for giant/64-bit, **w** for word/32-bit, **h** for halfword/16-bit, **b** for byte/8-bit).

**Step 7: Continue and Quit**

**continue**

**quit**