The gdb Debugger for C Programs

gcc -g -o *program* ...	Compile with debugging information
gdb [-d *sourcefiledir*] [-d *sourcefiledir*] ... *program* [*corefile*]	Run gdb from a shell
ESC x gdb [-d *sourcefiledir*] [-d *sourcefiledir*] ... *program* [*corefile*]	Run gdb from emacs

Miscellaneous	
quit	Exit gdb.
directory [*dir1*] [*dir2*] ...	Add directories *dir1*, *dir2*, ... to the list of directories searched for source files, or clear the directory list.
help [*cmd*]	Print a description of command *cmd*.

Listing the Source Code (or use emacs)	
list [[*file*:]*linenum1*[-*linenum2*]]	Print the source code lines numbered *linenum1* to *linenum2* in file *file*.
list [[*file*:]*fn*:][*linenum1*[-*linenum2*]]	Print the source code lines numbered *linenum1* to *linenum2* in function *fn* in file *file*.

Running the Program	
run [*arg1*],[*arg2*] …	Run the program with command-line arguments *arg1*, *arg2*, ...
set args *arg1 arg2 ...*	Set the program's command-line arguments to *arg1*, *arg2*, ...
show args	Print the program's command-line arguments.

Using Breakpoints	
info breakpoints	Print a list of all breakpoints.
break [*file*:]*linenum*	Set a breakpoint at line *linenum* in file *file*.
break [*file*:]*fn*	Set a breakpoint at the beginning of function *fn* in file *file*.
condition *bpnum expr*	Break at breakpoint *bpnum* only if expression *expr* is non-zero (TRUE).
commands [*bpnum*] *cmds*	Execute commands *cmds* whenever breakpoint *bpnum* is hit.
continue	Continue executing the program.
kill	Stop executing the program.
delete [*bpnum1*][,*bpnum2*]...	Delete breakpoints *bpnum1*, *bpnum2*, ..., or all breakpoints.
clear [[*file*:]*linenum*]	Clear the breakpoint at *linenum* in file *file*, or the current breakpoint.
clear [[*file*:]*fn*]	Clear the breakpoint at the beginning of function *fn* in file *file*, or the current breakpoint.
disable [*bpnum1*][,*bpnum2*]...	Disable breakpoints *bpnum1*, *bpnum2*, ..., or all breakpoints.
enable [*bpnum1*][,*bpnum2*]...	Enable breakpoints *bpnum1*, *bpnum2*, ..., or all breakpoints.

Stepping through the Program	
next	"Step over" the next line of the program.
step	"Step into" the next line of the program.
finish	"Step out" of the current function.

Examining Variables	
print *expr*	Print the value of expression *expr*.
print ['*file*'::]*var*	Print the value of variable *var* as defined in file *file*. (*File* is used to resolve static variables.)
print [*function*::]*var*	Print the value of variable *var* as defined in function *function*. (*Function* is used to resolve static variables.)
printf *format*, *expr1*, *expr2*, …	Print the values expressions *expr1*, *expr2*, ... using the specified *format* string.
whatis *var*	Print the type of variable *var*.
ptype *t*	Print the definition of type *t*.
info display	Print the display list.
display *expr*	At each break, print the value of expression *expr*.
undisplay *displaynum*	Remove *displaynum* from the display list.

Examining the Call Stack	
where	Print the call stack.
backtrace	Print the call stack.
frame	Print the top of the call stack.
up	Move the context toward the bottom of the call stack.
down	Move the context toward the top of the call stack.

Working with Signals	
info signals	Print a list of all signals that the operating system makes available.
handle *sig action1* [*action2* ...]	When gdb receives signal *sig*, it should perform actions *action1*, *action2*, ... Valid actions are nostop, stop, print, noprint, pass, and nopass.
signal *sig*	Send the program signal *sig*.