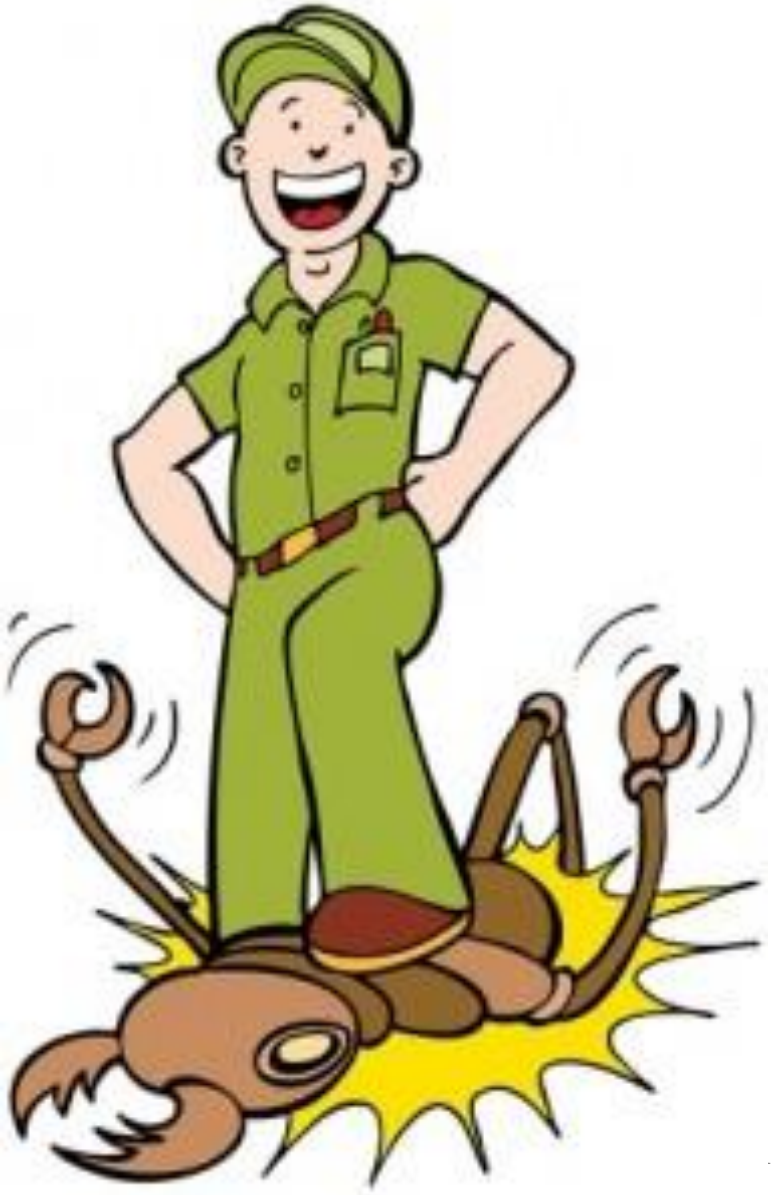
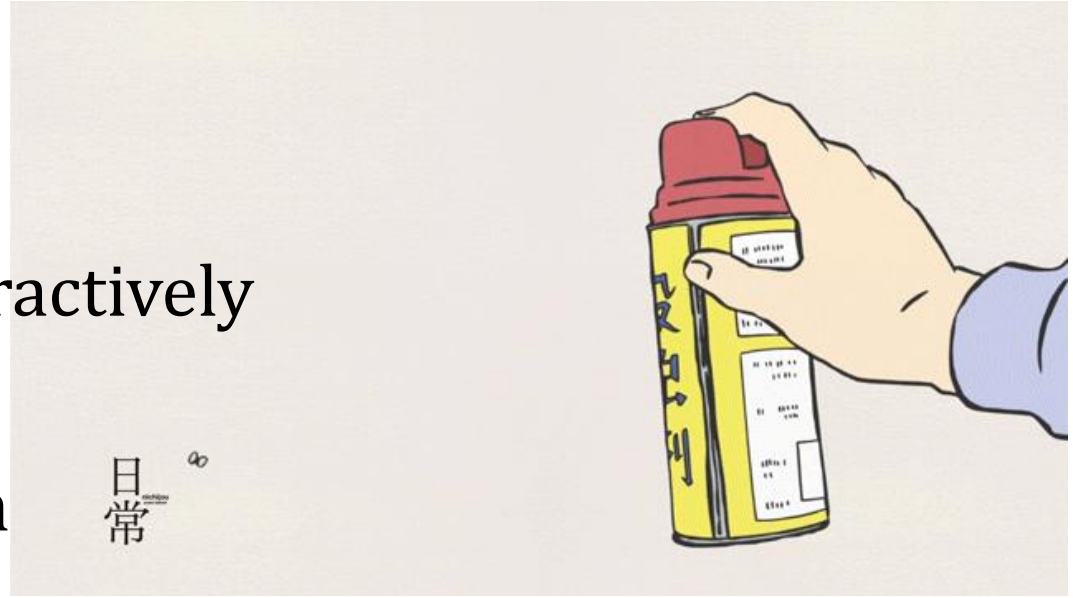
GDB



GNU DeBugger



Allows you to run your C program interactively

- Run up to a specific line or lines
- print out any C variable or expression
- Single step through your code
- Learn about the context of your code
  - Where were you called from?
  - What arguments were passed to you?
  - etc.

Running GDB

1. Your code MUST be compiled with the "-g" option
   - Adds debug information that gdb needs to your executable command
2. "Load" your code in the GDB environment
   - Run "gdb <yourcommandfile>"
   - gdb will start, print information about itself, load your code, and enter interactive mode

GDB Interactive Mode

- GDB writes a prompt…
  (gdb) _
- You type a gdb command and hit enter
  (gdb) print x
- GDB honors your command, then re-issues the prompt
  (gdb) print x
  $1 = 17
  (gdb) _

GDB Commands

CMD	Arguments	Description
break	*line* [if *cond*]	set a break point at the specified line
run		start program from beginning and run to first breakpoint
continue		run from current location to next breakpoint
print	*expression*	evaluate *expression*, and print the result
next		execute to next line if line is a function call, execute until that function returns
step		execute to next line if line is a function call, stop at first line of that function
where		Print out program call chain
info	*spec*	Print out information about this run of gdb
help	[*gdb cmd*]	print help for gdb
quit		Leave gdb

GDB Command Style

- GDB does not require the full command name
  - only enough to distinguish it from any other command
  - e.g. “p” is good enough for “print” because no other gdb commands start with “p”
- A null command (just enter) repeats the last command

```
(gdb) next
main.c:6 x=x+1;
(gdb)
main.c:7 y=y+1;
```

GDB Breakpoints

- List of line numbers at which GDB will stop and open up a prompt
- Set a break point with the break command
  (gdb) break 17
- Set a conditional breakpoint with the break/if command
  (gdb) break 21 if (x>10)
- List breakpoints using the info break command
  (gdb) info break
- Remove breakpoints using the clear command
  (gdb) clear 17

GDB Hints

- Open your code in a separate editor window before starting gdb
  - It's much easier to read your code in the editor than in gdb
- Invest some time getting comfortable with gdb
  - It will save you time over and over and over again!
  - ROI is enormous!
- No easy way to "back up" in gdb.
  - If you have gone too far, start again from the beginning. Either quit and restart gdb, or restart with the "run" command
- No easy way to change code and continue
  - If the code needs to be changed, need to quit, recompile, and restart gdb

GDB Demo

Resources

- Programming in C, Chapter 17
- On-line GDB manual (https://sourceware.org/gdb/current/onlinedocs/gdb/)
- Wikipedia: GNU Debugger (https://en.wikipedia.org/wiki/GNU_Debugger)