More of project structure
- gdb
Part 2 (Monday)

One annoying thing about C is having to manually free memory

.. malloc - - -

... free ---- or memory leaks

More modern langs use garbage collection to do this automatically (w/ a performance hit)

Part 2: build a small garbage collector!

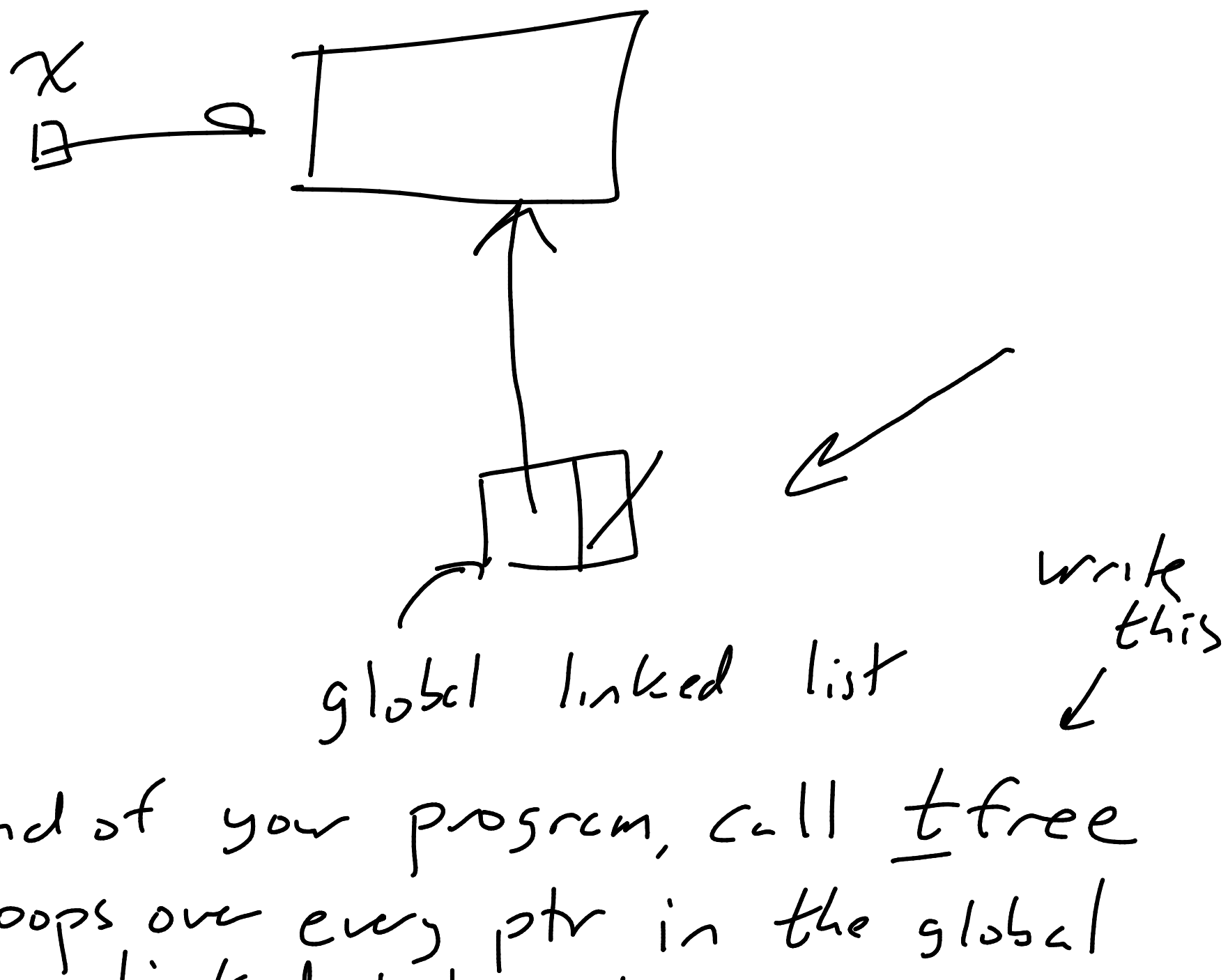
~~config time~~

Write a new function called talloc "track"

that does 2 things

1. calls malloc
2. adds a ptr to the new memory to a linked list

```
int *x = talloc(sizeof(int));
```



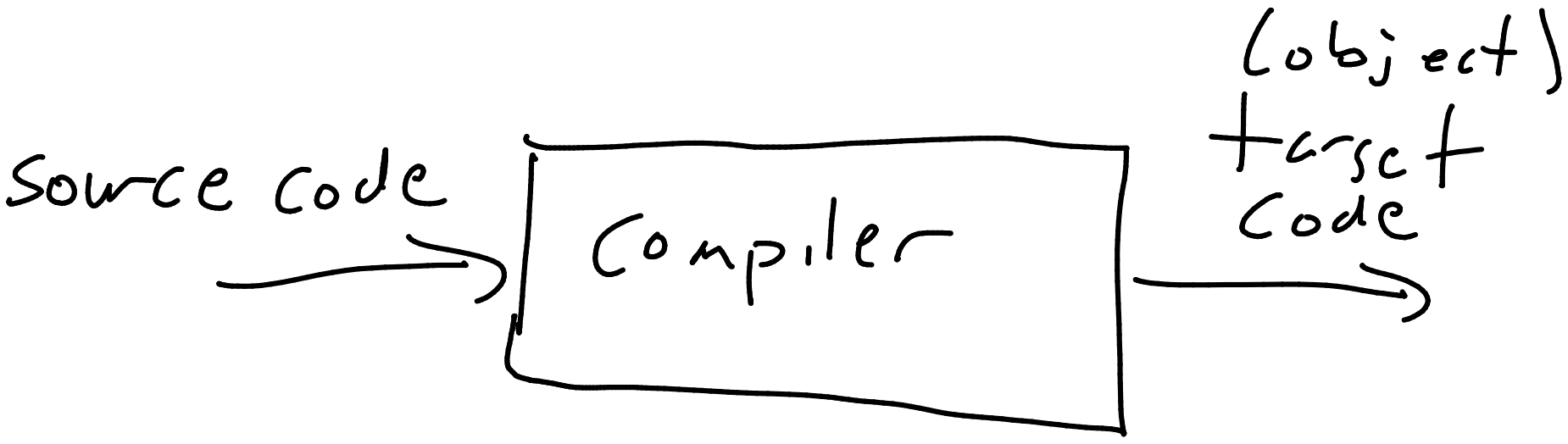
At end of your program, call tfree
- loops over every ptr in the global linked list and frees it

Rules of engagements

Compilers vs interpreters

- 2 diffent approaches for implementing a PL (approaches for translation)
- because if we are creating a new language, we need to rely on an existing language to make it happen (at the bottom, there's machine language)

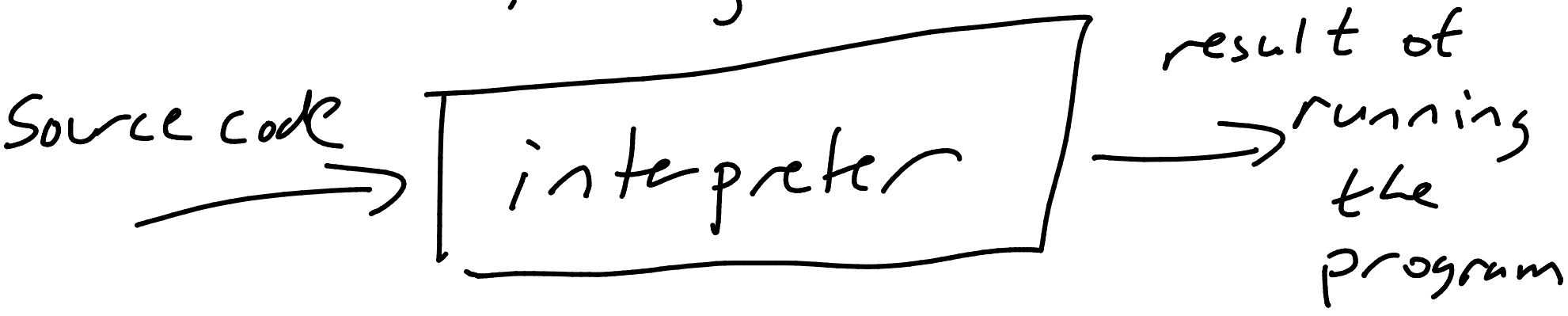
A compiler is a program that translates a program from one language to another (usually lower level)



C program → clang → machine language

An interpreter is a program that reads and directly executes another program.

- it "pretends" to understand the source code.
- there's no translated code that comes out, it just runs it



Python program → fictional Python interpreter that you think you're using → program output