

1. (a) Name the two parts of a compiler's front end.

**Lexing (or lexical analysis) and parsing (or syntactic analysis)**

- (b) Name the two parts of a compiler's back end.

**Optimization (or machine-independent optimization) and code generation (or machine-dependent optimization).**

- (c) What are the two outputs of the front end?

**Abstract syntax tree and symbol table.**

2. Intermediate representations (IRs) often look like simplified machine languages. Give two reasons why compilers use IRs internally instead of machine language.

**Simplifies machine-independent optimization  
Easier to port compiler to new target machine**

3. Name four of the items in an activation record.

**(Four of:)      Function arguments  
                  Return address  
                  Dynamic link  
                  Saved registers  
                  Local variables**

4. Stop-and-copy garbage collection is much more popular than mark-and-sweep. Explain why.

**Stop-and-copy is usually much more efficient than mark-and-sweep. Mark and sweep requires a pass over the entire heap (to find the unmarked cells), while stop-and-copy requires only a traversal of the reachable cells, which are often a small fraction of the entire heap space.**