Problem set 1

1.

gcc 4.8.2
flex 2.5.35
bison 3.0.2

2.

a)
An interpreter translates and executes one statement at a time. When an error occurs the program stops. Analyzing the code also takes less time. The execution time on interpreters is normally slower compared to a compiler.

A compiler analyses the whole program and then translates it to machine code. Error messages is shown after the whole program is analyzed.
b)
Compilers converts source code from a high level programing language to a low level language. E.g C compiles to Assembly. Translators translates one programming language to another programming language. But instead of translating from a high level language it can also translate one high level language to another high level language.

3.

The compile stages is divided into two main groups frontend and backend.

Frontend

- Lexical analysis

Analyzing a stream of characters from the source code and make tokens.

- Syntax analysis

Generated tokens is grouped together from the language grammar.

- Semantic analysis

The outputted parse tree is analyzed for semantic errors. E.g assign an int value to a char variable. It is syntactically correct but semantically wrong.

- Intermediate code generation

Intermediate code is generated from the source code. The code is often represented in a structure called three-address code.

Backend

- IR optimization

Optimization of the code is done by removing unused variables, remove unreachable code, remove statements that is not changed inside loops and so on.

- Object code generation

The output from this stage is mainly machine code or assembly code. Generated code from this stage will match the targeted hardware.

- Object code optimization

In this step hardware specific optimizations are taken into account. The targeted hardware may have optimized instructions for special cases.