1. What are two parts of compiler? Explain briefly.

Compiler has 2 parts namely, Analysis and Synthesis stage. Analysis part of the compiler breaks the source program into a stream of tokens and creates a language independent intermediate representation of the program. Synthesis part generates the desired target program from intermediate representation.

1. Illustrate diagrammatically how a language is processed.

Source Program

Preprocessor

Modified Source Program

Compiler

Target Assembly Program

Assembler

Relocatable machine code

Library files

Relocatable object files

Linker/Loader

Target machine code

1. Explain the various errors encountered in different phases of the compiler

**Errors in Lexical phase**

**Misspelt keyword** can be recognized as identifiers

fi(a==f(x))---

fi matches to the pattern of identifier

**Invalid token, invalid character**

If lexical analyzer is unable to proceed because none of the patterns for the tokens match with the input lexeme, it tries to panic mode recovery

Possible recovery options would be

* Delete one character from the remaining input
* Insert a misspelt character in the remaining input
* Replaces a character by another character
* Transpose two adjacent characters

**Identifier too long**: As the compiler writer you must decide what the maximum length of an identifier is, and issue an error when that length is exceeded.

**Errors in Syntax Analysis Phase**

**Error Recovery Methods**

**Panic mode:**

Skipping the tokens until it encounters a synchronizing token. For example, while reading if it detects erroneous input it just skips until it gets the fresh statement which is correct

**Phrase level:**

When a parser encounters an error, it tries to take corrective measures so that the rest of inputs of statement allow the parser to parse ahead. For example, inserting a missing semicolon, replacing comma with a semicolon etc.

Examples:

int a 5; 🡪 insert = between a & 5

int a:b:c; 🡪 replace each colon by ;

**Error productions:**

Some common errors are known to the compiler designers that may occur in the code. In addition, the designers can create augmented grammar to be used, as productions that generate erroneous constructs when these errors are encountered.

Example: Include production with ; and also with ,

If production with , is encountered throw an error

**Global Error Correction:**

The parser considers the program in hand as a whole and tries to figure out what the program is intended to do and tries to find out a closest match for it, which is error-free with minimal changes.

**Errors in Semantic Analysis Phase**

These errors are a result of incompatible value assignment. The semantic errors that the semantic analyzer is expected to recognize are:

• Type mismatch.  
• Undeclared variable.  
• Reserved identifier misuse.  
• Multiple declaration of variable in a scope.  
• Accessing an out of scope variable.  
• Actual and formal parameter mismatch.

In code optimization, errors occur when the result is affected by the optimization.

In code generation, it shows error when code is missing etc.

1. asdsd