Transducer or E-programs:
replace, count, concordance
type checking, compiling, translation
Lex, yacc
① Transducer operation ② Shift parser
③ E-programs ④ Lex, yacc, Flex, Bison

Data Types:
Integers
① Decimal ② Hexadecimal
③ Octal ④ Binary
⑤ Big integer, big number (Bignum)
⑥ Fixed-precision ⑦ Floating-point
⑧ Complex ⑨ Rational
⑩ Fixed-point ⑪ Arbitrary-precision
Integer size
8-bit
16-bit
32-bit
64-bit

Sequence Data Types
① Array ② Linked list
③ Dynamic array ④ Queue
⑤ Hash table ⑥ Stack
⑦ Tree ⑧ Graph
⑨ Heap ⑩ Trie
⑪ Suffix array

Strings
① Character ② Unicode ③ Binary

Composite data types
① Record data types ② Class ③ Class without data types
④ Enumerations ⑤ Array
⑥ Tuple ⑦ Struct
⑧ Dictionary ⑨ Hashmap
⑩ Set ⑪ Linked list
⑫ Queue ⑬ Stack
⑭ Tree ⑮ Heap
⑯ Graph ⑰ Trie
⑱ Suffix array

Generics of Array
① Int ② Char ③ Bool ④ Int[N]
Char[N]-Char ⑤ Int[N]
Char[N]-Int ⑥ Int[N]-Int[N]
⑦ Int[N]-Int[N]-Int[N]
Int[N]-Int[N]-Int[N]-Int[N]
⑧ Int[N]-Int[N]-Int[N]-Int[N]-Int[N]
Int[N]-Int[N]-Int[N]-Int[N]-Int[N]-Int[N]
Int[N]-Int[N]-Int[N]-Int[N]-Int[N]-Int[N]-Int[N]
Int[N]-Int[N]-Int[N]-Int[N]-Int[N]-Int[N]-Int[N]-Int[N]
Int[N]-Int[N]-Int[N]-Int[N]-Int[N]-Int[N]-Int[N]-Int[N]-Int[N]
Int[N]-Int[N]-Int[N]-Int[N]-Int[N]-Int[N]-Int[N]-Int[N]-Int[N]-Int[N]
Int[N]-Int[N]-Int[N]-Int[N]-Int[N]-Int[N]-Int[N]-Int[N]-Int[N]-Int[N]-Int[N]
Int[N]-Int[N]-Int[N]-Int[N]-Int[N]-Int[N]-Int[N]-Int[N]-Int[N]-Int[N]-Int[N]-Int[N]
Int[N]-Int[N]-Int[N]-Int[N]-Int[N]-Int[N]-Int[N]-Int[N]-Int[N]-Int[N]-Int[N]-Int[N]-Int[N]
Int[N]-Int[N]-Int[N]-Int[N]-Int[N]-Int[N]-Int[N]-Int[N]-Int[N]-Int[N]-Int[N]-Int[N]-Int[N]-Int[N]
Int[N]-Int[N]-Int[N]-Int[N]-Int[N]-Int[N]-Int[N]-Int[N]-Int[N]-Int[N]-Int[N]-Int[N]-Int[N]-Int[N]-Int[N]
Int[N]-Int[N]-Int[N]-Int[N]-Int[N]-Int[N]-Int[N]-Int[N]-Int[N]-Int[N]-Int[N]-Int[N]-Int[N]-Int[N]-Int[N]-Int[N]
Int[N]-Int[N]-Int[N]-Int[N]-Int[N]-Int[N]-Int[N]-Int[N]-Int[N]-Int[N]-Int[N]-Int[N]-Int[N]-Int[N]-Int[N]-Int[N]-Int[N]
Int[N]-Int[N]-Int[N]-Int[N]-Int[N]-Int[N]-Int[N]-Int[N]-Int[N]-Int[N]-Int[N]-Int[N]-Int[N]-Int[N]-Int[N]-Int[N]-Int[N]-Int[N]
Int[N]-Int[N]-Int[N]-Int[N]-Int[N]-Int[N]-Int[N]-Int[N]-Int[N]-Int[N]-Int[N]-Int[N]-Int[N]-Int[N]-Int[N]-Int[N]-Int[N]-Int[N]-Int[N]
Int[N]-Int[N]-Int[N]-Int[N]-Int[N]-Int[N]-Int[N]-Int[N]-Int[N]-Int[N]-Int[N]-Int[N]-Int[N]-Int[N]-Int[N]-Int[N]-Int[N]-Int[N]-Int[N]-Int[N]
Int[N]-Int[N]
Int[N]-Int[N]
Int[N]-Int[N]
Int[N]-Int[N]
Int[N]-Int[N]-Int[N]-Int[N]-Int[N]-Int[N]-Int[N]-Int[N]-Int[N]-Int[N]-Int[N]-Int[N]-Int[N]-Int[N]-Int[N]-Int[N]-Int[N]-Int[N]-Int[N]-