Q1. Self-organizing list based symbol tables may show better performance due to

(A) Locality of input program
(B) Locality of compiler
(C) Both locality of input program and compiler
(D) None of the given options

Ans: A

Q2. Which of the following is NOT likely to be kept in a symbol table?

(A) Name
(B) Location
(C) Scope
(D) None of the other options

Ans: D

Q3. Which of the following phases of compiler does NOT use symbol table?

(A) Semantic analysis
(B) Code generation
(C) Code optimization
(D) None of the given options

Ans: D

Q4. One symbol table per scope is suited for

(A) Single-pass compilers
(B) Multi-pass compilers
(C) Both single- and multi-pass compilers
(D) None of the given options

Ans: A

Q5. If two types have same name they can be

(A) Name equivalent
(B) Structurally equivalent
(C) Both name and structurally equivalent
(D) May not be name equivalent

Ans: C

Q6. Symbol table data is filled by

(A) Lexical analyzer
(B) Parser
(C) Both lexical analyzer and parser
(D) Neither lexical analyzer nor parser

Ans: C

Q7. Most frequent operation on a symbol table is

(A) Insert
(B) Delete
(C) Modify
(D) Lookup

Ans: D

Q8. Motivation behind using self-organizing list for symbol table is

(A) Ease of implementation

(B) Program locality
(C) Insertion of symbols
(D) None of the other options

Ans: B

Q9. To minimize access time, symbol table should be organized as
(A) Linear table
(B) Tree
(C) Hash Table
(D) Circular list

Ans: C

Q10. Activation record stores
(A) Parameters
(B) Local variables
(C) Parameters and local variables
(D) Parameters, local variables and code for procedures

Ans: C