Q1. For the rule S → L := E, if L is a single variable, L.place is equal to

(A) Null
(B) Some value
(C) Constant
(D) None of the other options

Ans: A

Q2. For the rule B → B1 and B2, the operation "B1.false = B.false" requires two passes as

(A) B1.false is not known
(B) B.false is not known
(C) Both B1.false and B.false are unknown
(D) None of the other options

Ans: B

Q3. For Boolean variable B, B.truelist contains

(A) List of locations at which B is true
(B) List of locations to jump to if B is true
(C) List of locations at which B is true and the locations to branch to
(D) None of the other options

Ans: A

Q4. In the rule B → B1 or MB2, the nonterminal M is used to remember the start address of

(A) B
(B) B1
(C) Both B1 and B
(D) None of the other options

Ans: D

Q5. When code is generated for "a < b and c > d", the locations left for backpatching are

(A) falselist of a < b
(B) falselist of a < b and falselist of c > d
(C) falselist of a < b, falselist of c < d, truelist of c < d
(D) truelist of a < b, falselist of a < b, truelist of c < d, falselist of c < d

Ans: C

Q6. In the rule S → if B then M S N else M S, N is used to generate a jump after

(A) then-part
(B) else-part
(C) both then- and else-part
(D) None of the other options

Ans: A

Q7. In three-address code, arrays are

(A) Not supported
(B) One dimensional
(C) More than one dimensional
(D) Supported via pointers

Ans: B

Q8. In the rule S →if B then M S1, M holds the start address for

(A) S1
(B) S
(C) B
(D) None of the other options

Ans: A

Q9. For three address code generation of "B → B1 or M B2", M.quad is used to backpatch

(A) B1.truelist
(B) B1.falselist
(C) B2.truelist
(D) B2.falselist

Ans: B

Q10. For three address code generation of rule "S → while M1 B do M2 S1", B.falselist is backpatched with

(A) M1.quad
(B) M2.quad
(C) Cannot be backpatched at this point
(D) None of the other options

Ans: C