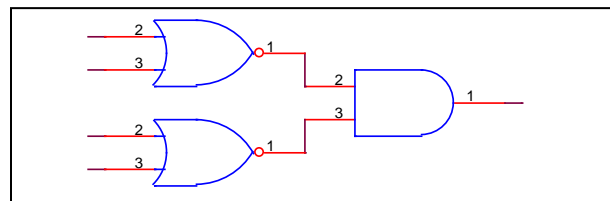
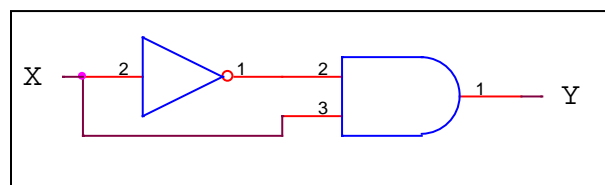


LEARNING OUTCOME #2: “an ability to analyze and design combinational logic circuits.”**Multiple Choice – select the single most appropriate response for each question.****Note that “none of the above” MAY be a VALID ANSWER.**

- If the function $F(X,Y,Z)$ is represented by the **ON SET** $\Sigma_{X,Y,Z}(0,3,5,6)$, then the **complement** of this function $F'(X,Y,Z)$ is represented by the **ON SET**:
 - $\Sigma_{X,Y,Z}(0,3,5,6)$
 - $\Sigma_{X,Y,Z}(1,2,4,7)$
 - $\Sigma_{X,Y,Z}(1,2,4,6)$
 - $\Sigma_{X,Y,Z}(1,3,5,7)$
 - none of the above
- If the function $F(X,Y,Z)$ is represented by the **ON SET** $\Sigma_{X,Y,Z}(0,3,5,6)$, then the **dual** of this function $F^D(X,Y,Z)$ is represented by the **ON SET**:
 - $\Sigma_{X,Y,Z}(0,3,5,6)$
 - $\Sigma_{X,Y,Z}(1,2,4,7)$
 - $\Sigma_{X,Y,Z}(1,2,4,6)$
 - $\Sigma_{X,Y,Z}(1,3,5,7)$
 - none of the above
- The **XOR property** listed below that is **NOT** true is:
 - $X \oplus 0 = X$
 - $X \oplus 1 = X'$
 - $X \oplus X = X$
 - $X \oplus X' = 1$
 - none of the above
- A circuit consisting of a level of **NOR gates** followed by a level of **AND gates** is **logically equivalent** to:
 - a multi-input OR gate
 - a multi-input AND gate
 - a multi-input NOR gate
 - a multi-input NAND gate
 - none of the above



- The circuit shown exhibits the **following type of hazard** when its input, X , transitions from **low-to-high**:
 - a static-zero hazard
 - a static-one hazard
 - a dynamic hazard
 - a consensus hazard
 - none of the above



The following K-map applies to questions 6 through 11:

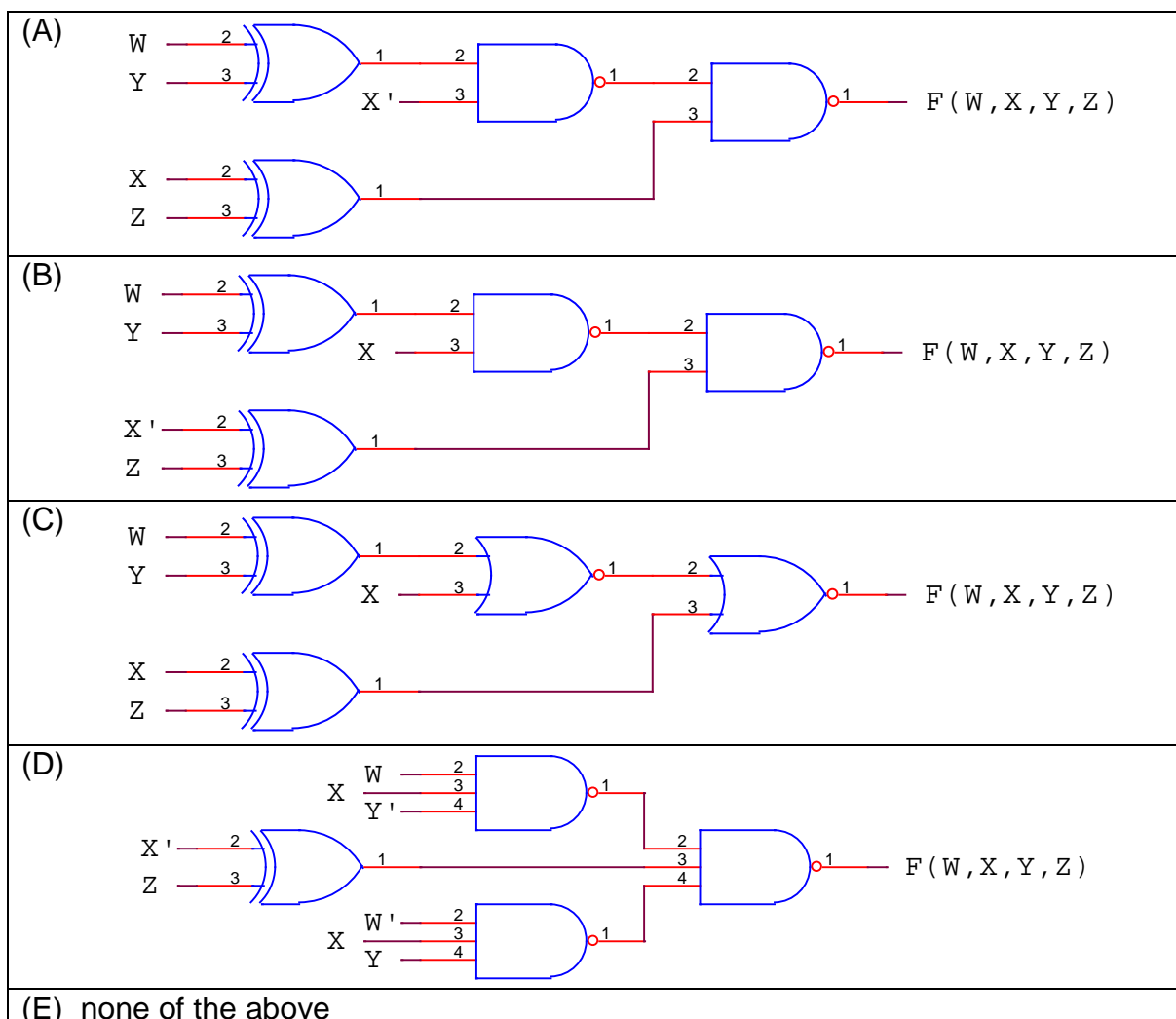
	X'		X	
Z'	0	1	0	0
Z	1	0	d	1
	Y'	Y		Y'

6. The **cost** of a **minimal sum of products** realization of this function (assuming **both true and complemented variables** are available) would be:
 (A) 10 (B) 11 (C) 12 (D) 13 (E) none of the above
7. The **cost** of a **minimal product of sums** realization of this function (assuming **both true and complemented variables** are available) would be:
 (A) 10 (B) 11 (C) 12 (D) 13 (E) none of the above
8. Assuming the availability of **only true** input variables, the **fewest number of 2-input NAND gates** that are needed to realize this function is:
 (A) 5 (B) 6 (C) 7 (D) 8 (E) none of the above
9. Assuming the availability of **only true** input variables, the **fewest number of 2-input NOR gates** that are needed to realize this function is:
 (A) 5 (B) 6 (C) 7 (D) 8 (E) none of the above
10. Assuming the availability of **only true** input variables, the **fewest number of 2-input open-drain NAND gates** that are needed to realize this function is:
 (A) 5 (B) 6 (C) 7 (D) 8 (E) none of the above
11. Assuming the availability of **only true** input variables, the **number of pull-up resistors** required to realize this function using **2-input open-drain NAND gates** is:
 (A) 1 (B) 2 (C) 3 (D) 4 (E) none of the above

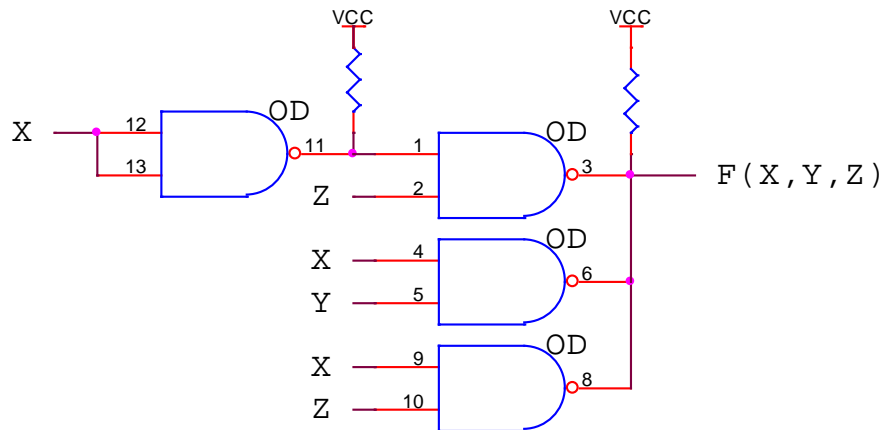
The following K-map applies to question 12:

	W'		W		
Y'	0	d	1	0	Z'
	1	0	1	d	Z
Y	d	1	0	1	Z
	0	1	1	0	Z'
	X'	X		X'	

12. Assuming the availability of **both true and complemented** variables, the **simplest** (lowest cost) realization of this function is depicted by the following circuit:

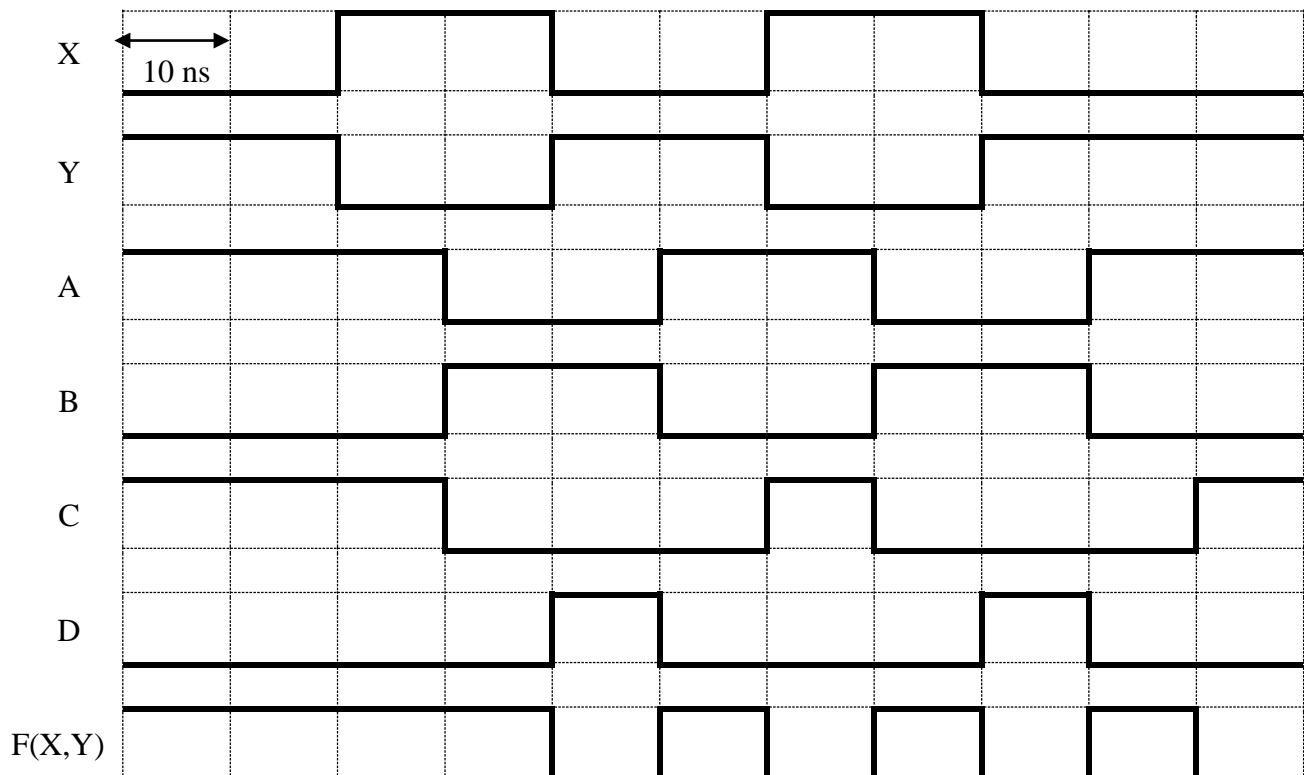
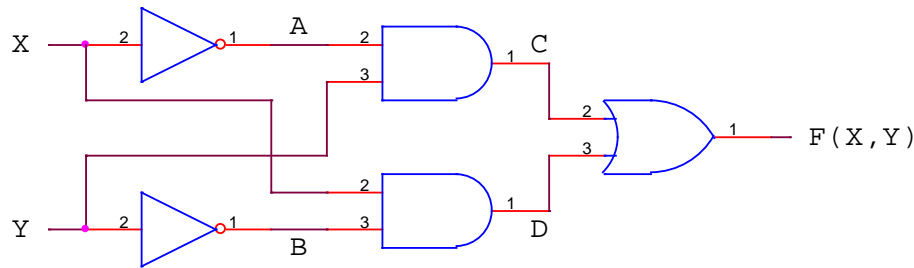


The following circuit applies to questions 13 through 16:



13. Expressed in a **minimum sum-of-products** form, the function realized by this circuit is:
- $X \cdot Y + X \cdot Z + X' \cdot Z$
 - $X' \cdot Z' + Y' \cdot Z'$
 - $Y \cdot Z' + Y' \cdot Z' + X' \cdot Y'$
 - $Z + X \cdot Y$
 - none of the above
14. The **ON-SET** of the function realized by this circuit is:
- $\sum_{X,Y,Z}(0,2,4)$
 - $\sum_{X,Y,Z}(1,3,5,6,7)$
 - $\sum_{X,Y,Z}(3,5,7)$
 - $\sum_{X,Y,Z}(0,1,2,4,6)$
 - none of the above
15. Expressed in a **minimum product-of-sums** form, the function realized by this circuit is:
- $(X+Z) \cdot (Y+Z)$
 - $(X'+Y') \cdot (X+Y) \cdot (Y+Z')$
 - $Z' \cdot (X'+Y')$
 - $(X'+Z') \cdot (Y'+Z')$
 - none of the above
16. Assuming the availability of **only true** variables, realization of this same function using **only 2-input NOR gates** would require:
- two** (2-input NOR) gates
 - three** (2-input NOR) gates
 - four** (2-input NOR) gates
 - five** (2-input NOR) gates
 - none of the above

The following circuit and timing chart apply to questions 17 and 18:



17. Steady-state (static) analysis of the function realized by this circuit predicts that the output $F(X, Y)$ should:

- (A) always be low
- (B) always be high
- (C) be equal to the input X
- (D) be equal to the input Y
- (E) none of the above

18. The dynamic behavior of the circuit (depicted in the timing chart) does **NOT** match the steady-state analysis of the function because:

- (A) the circuit does not contain a complete sum
- (B) a consensus term needs to be added
- (C) more than one input changes simultaneously
- (D) all of the above
- (E) none of the above

The following excerpt from an **ispLever Reduced Equation Report** applies to questions 19 through 21.

Title: 8-to-3 Priority Encoder

P-Terms	Fan-in	Fan-out	Type	Name (attributes)
4/4	7	1	Pin	E0
4/3	6	1	Pin	E1
4/1	4	1	Pin	E2
8/1	8	1	Pin	GS
=====				
20/9	Best P-Term Total: 9			
	Total Pins: 12			
	Total Nodes: 0			
	Average P-Term/Output: 2			

ispLEVER operators:
 AND - & OR - #
 NOT - ! XOR - \$

Positive-Polarity Equations:

E0 = (!I6 & !I4 & !I2 & I1 # !I6 & !I4 & I3 # !I6 & I5 # I7);

E1 = (!I5 & !I4 & I2 # !I5 & !I4 & I3 # I6 # I7);

E2 = (I4 # I5 # I6 # I7);

GS = (I1 # I0 # I2 # I3 # I4 # I5 # I6 # I7);

Reverse-Polarity Equations:

!E0 = (!I7 & !I5 & !I3 & !I1 # !I7 & !I5 & !I3 & I2 # !I7 & !I5 & I4 # !I7 & I6);

!E1 = (!I7 & !I6 & !I3 & !I2 # !I7 & !I6 & I4 # !I7 & !I6 & I5);

!E2 = (!I7 & !I6 & !I5 & !I4);

!GS = (!I7 & !I6 & !I5 & !I4 & !I3 & !I2 & !I1 & !I0);

19. The **Best P-Term Total** shown in the report indicates:

- (A) the minimum number of pins needed to implement the logic function
- (B) the minimum number of macrocells needed to implement the logic function
- (C) the minimum number of OR gates needed to implement the logic function
- (D) the minimum number of AND gates needed to implement the logic function
- (E) none of the above

20. The **number of P-Terms** needed to realize the **!E0** (reverse-polarity) equation is:

- (A) 1 (B) 2 (C) 4 (D) 8 (E) none of these

21. The **number of P-Terms** needed to realize the **GS** (positive-polarity) equation is:

- (A) 1 (B) 2 (C) 4 (D) 8 (E) none of these

The following Verilog module applies to questions 22 through 24:

```
module diff_pri(A,B,C,D,EN,E0,E1,GS);
    input wire A,B,C,D,EN;
    output wire E0, E1;
    output wire GS;

    reg [2:0] EGS;

    always @ (A,B,C,D) begin
        casez ({A,B,C,D})
            4'b0000: EGS = 3'b000;
            4'b0001: EGS = 3'b111;
            4'b001?: EGS = 3'b101;
            4'b01??: EGS = 3'b011;
            4'b1???: EGS = 3'b001;
        endcase
    end

    assign E1 = EN ? EGS[2]:1'bz;
    assign E0 = EN ? EGS[1]:1'bz;
    assign GS = EGS[0];

endmodule
```

22. The **highest priority input** is:

- (A) A
- (B) B
- (C) C
- (D) D
- (E) none of the above

23. If input **A is asserted** and input **EN is negated**, the outputs will be:

- (A) E1=0, E0=0, GS=0
- (B) E1=0, E0=0, GS=1
- (C) E1=Hi-Z, E0=Hi-Z, GS=1
- (D) E1=Hi-Z, E0=Hi-Z, GS=Hi-Z
- (E) none of the above

24. When inputs **B** and **C** are **asserted simultaneously** (and EN is asserted), the **encoded output** will be:

- (A) 00
- (B) 01
- (C) 10
- (D) 11
- (E) none of the above

The following Verilog module applies to questions 25 through 27:

```
module bigmux(EN,S,A,B,C,D,Y);
    input wire EN;
    input wire [1:0] S;
    input wire [7:0] A,B,C,D;
    output tri [7:0] Y;

    wire [7:0] Ytmp,SEL1,SEL0;

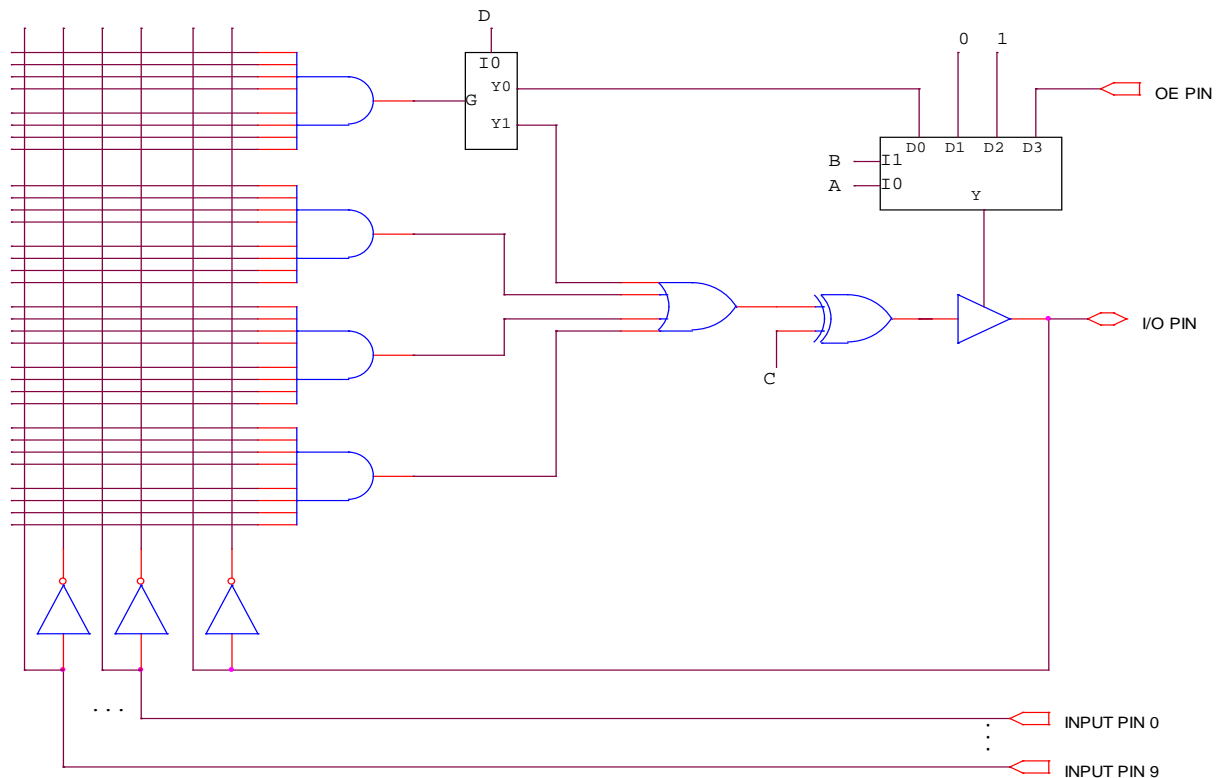
    assign SEL1 = {8{S[1]}};
    assign SEL0 = {8{S[0]}};

    assign Ytmp = (~SEL1 & ~SEL0 & A) | (~SEL1 & SEL0 & B) |
                  (SEL1 & ~SEL0 & C) | (SEL1 & SEL0 & D);

    assign Y = EN ? Ytmp : 8'bzzzzzzzz;
endmodule
```

25. The **number of equations** generated by this program (that would be burned into a PLD that realized this design) is:
- (A) 2
 - (B) 8
 - (C) 9
 - (D) 16
 - (E) none of the above
26. When **EN=0**, **S1=1**, and **S0=1**, bits 0 through 7 of output **Y** will:
- (A) all be Hi-Z
 - (B) all be zero
 - (C) all be one
 - (D) be equal to bits 0 through 7 of input **D**
 - (E) none of the above
27. When **EN=1**, **S1=1**, and **S0=1**, bits 0 through 7 of output **Y** will:
- (A) all be Hi-Z
 - (B) all be zero
 - (C) all be one
 - (D) be equal to bits 0 through 7 of input **D**
 - (E) none of the above

The **Macrocell Reference Figure** below applies to questions 28 through 30.



28. If **D=1**, the **maximum number of product terms** that can be implemented by each macrocell is:
 (A) 3 (B) 4 (C) 8 (D) 32 (E) none of these
29. To realize the **ON SET** of a function with an **active low** output, the following settings should be used:
 (A) **C=0, B=0, A=1**
 (B) **C=0, B=1, A=0**
 (C) **C=1, B=0, A=1**
 (D) **C=1, B=1, A=0**
 (E) none of the above
30. The purpose of setting **B=0** and **A=1** is to:
 (A) force the macrocell to use the PoS form of the logic equations
 (B) enable one of the AND gates to be the source for the tri-state output enable
 (C) make the OE pin the source for the tri-state enable
 (D) enable the I/O pin to be used as an extra input
 (E) none of the above

1-B, 2-A, 3-C, 4-C, 5-A, 6-A, 7-D, 8-D, 9-E, 10-A, 11-C, 12-B, 13-B, 14-A, 15-C, 16-C, 17-B, 18-C, 19-D, 20-C, 21-D, 22-A, 23-C, 24-B, 25-D, 26-A, 27-D, 28-B, 29-D, 30-D

Answer key:

Answer key:

1-B, 2-A, 3-C, 4-C, 5-A, 6-A, 7-D, 8-D, 9-E, 10-A, 11-C, 12-B, 13-B, 14-A, 15-C,
16-C, 17-B, 18-C, 19-D, 20-C, 21-D, 22-A, 23-C, 24-B, 25-D, 26-A, 27-D, 28-B,
29-D, 30-D