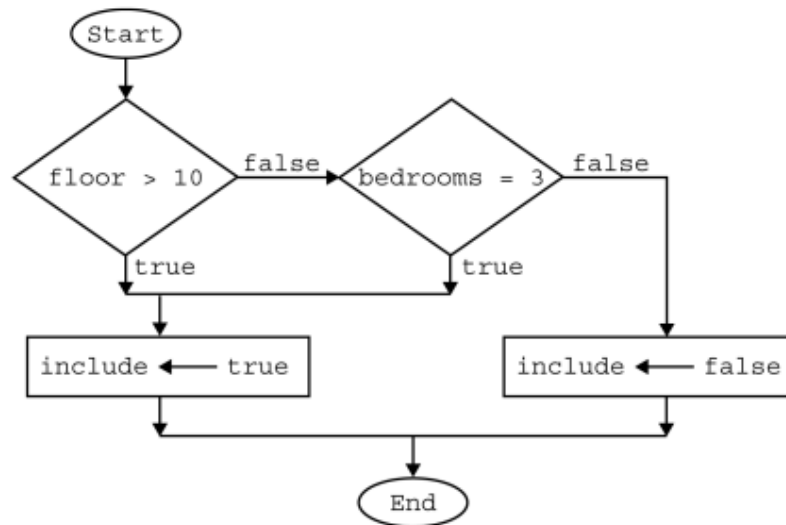


Name _____ Period _____

Skill 29.01 Exercise 1

Block	Explanation
Oval ○	The start or end of the algorithm
Diamond ◇	A conditional or decision step, where execution proceeds to the side labeled <i>true</i> if the condition is true and to the side labeled <i>false</i> otherwise
Rectangle □	One or more processing steps, such as a statement that assigns a value to a variable



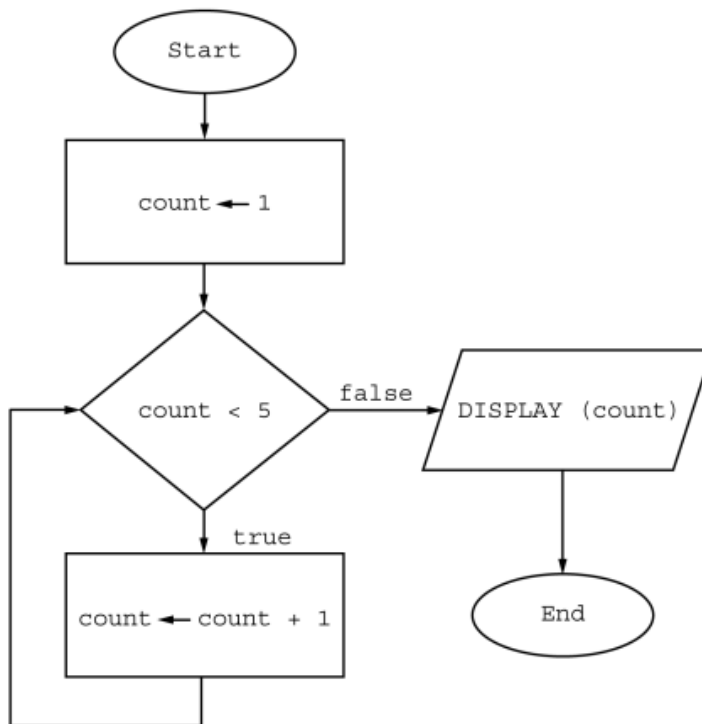
Which of the following statements is equivalent to the algorithm in the flowchart?

- (A) `include ← (floor > 10) OR (bedrooms = 3)`
- (B) `include ← (floor > 10) AND (bedrooms = 3)`
- (C) `include ← (floor ≤ 10) OR (bedrooms = 3)`
- (D) `include ← (floor ≤ 10) AND (bedrooms = 3)`

Name _____ Period _____

Skill 29.01 Exercise 2

Block	Explanation
Oval ○	The start or end of the algorithm
Rectangle □	One or more processing steps, such as a statement that assigns a value to a variable
Diamond ◇	A conditional or decision step, where execution proceeds to the side labeled <code>true</code> if the condition is true and to the side labeled <code>false</code> otherwise
Parallelogram ▱	Displays a message



What is displayed as a result of executing the algorithm in the flowchart?

- (A) 5
- (B) 15
- (C) 1 2 3 4
- (D) 1 2 3 4 5

Name _____ Period _____

Skill 29.01 Exercise 3

Central High School keeps a database of information about each student, including the numeric variables `numberOfAbsences` and `gradePointAverage`. The expression below is used to determine whether a student is eligible to receive an academic award.

$(\text{numberOfAbsences} \leq 5) \text{ AND } (\text{gradePointAverage} > 3.5)$

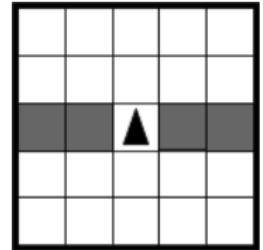
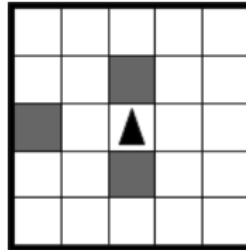
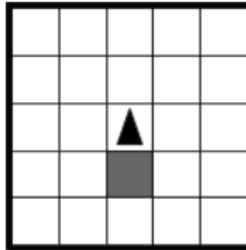
Draw a flowchart to represent the statement above. If the conditions above are met, the variable *academicAward* is true, otherwise it is false.

Name _____ Period _____

Skill 29.02 Exercises 1 thru 3

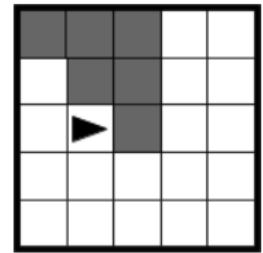
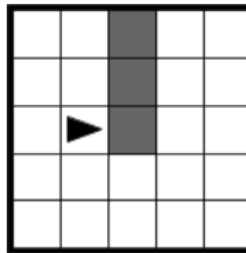
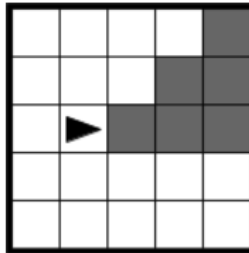
Basic If-Statements

```
ROTATE_LEFT ()
IF (CAN_MOVE (left))
{
    ROTATE_LEFT ()
}
MOVE_FORWARD ()
MOVE_FORWARD ()
```

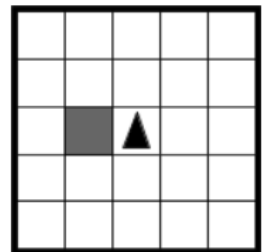
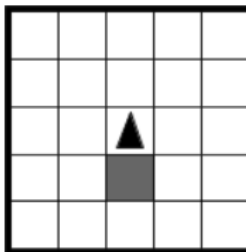
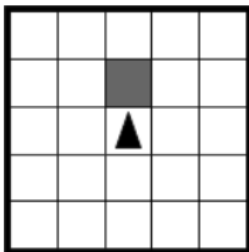


Sequential If-Statements

```
ROTATE_LEFT ()
IF (CAN_MOVE (forward))
{
    MOVE_FORWARD ()
}
ROTATE_RIGHT ()
IF (CAN_MOVE (forward))
{
    MOVE_FORWARD ()
}
ROTATE_LEFT ()
IF (CAN_MOVE (forward))
{
    MOVE_FORWARD ()
}
```



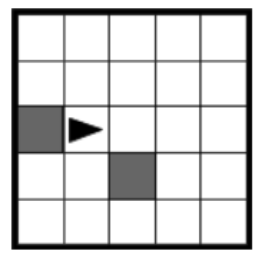
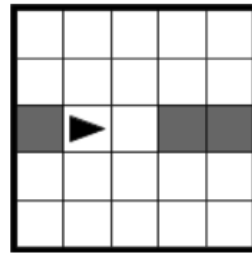
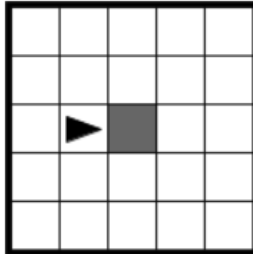
```
IF (CAN_MOVE ( left ))
{
    ROTATE_LEFT ()
    MOVE_FORWARD ()
}
IF (CAN_MOVE ( left ))
{
    ROTATE_LEFT ()
    MOVE_FORWARD ()
}
IF (CAN_MOVE ( left ))
{
    ROTATE_LEFT ()
    MOVE_FORWARD ()
}
```



Name _____ Period _____

Nested If-Statement

```
IF (CAN_MOVE (forward))  
{  
    MOVE_FORWARD ()  
    IF (CAN_MOVE (left))  
    {  
        ROTATE_LEFT ()  
        IF (CAN_MOVE (right))  
        {  
            ROTATE_RIGHT ()  
        }  
    }  
}  
MOVE_FORWARD ()
```



Skill 29.03 Exercise 1

Write a function called *timeToBuy*. You function should accept a parameter called *sale*. If *sale* is true, return “Time to buy!”, otherwise return “Wait for a sale!”.

Write a function called *canDrive*. The function you write should accept a parameter called *age*. In the body of the function return *true* if *age* is 16 years or older, otherwise return false.

Name _____ Period _____

Skill 29.04 Exercise 1

Write a function called *sortWords*. *Sort words* should accept two parameters, *word1* and *word2*. In the body of the function, check if the words are sorted. If the words are not sorted, assign *word1* and *word2* to their correct values. Your function should return a string that summarizes the values of *word1* and *word2*. Consider the examples below,

Values before call		Values after call	
word1	cat	word1	bat
word2	bat	word2	cat

Indicate two calls, along with corresponding `console.log` statements, that could be used to demonstrate that your function works properly.

Indicate the output of the calls above.