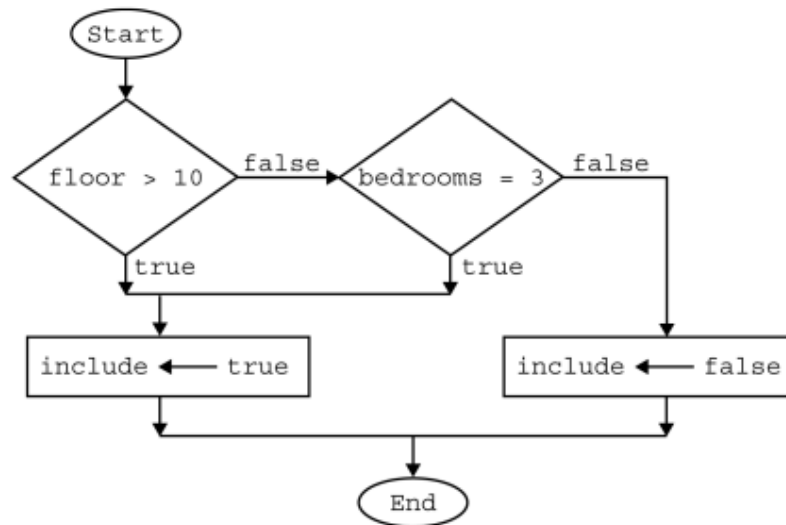


Name _____ Period _____

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



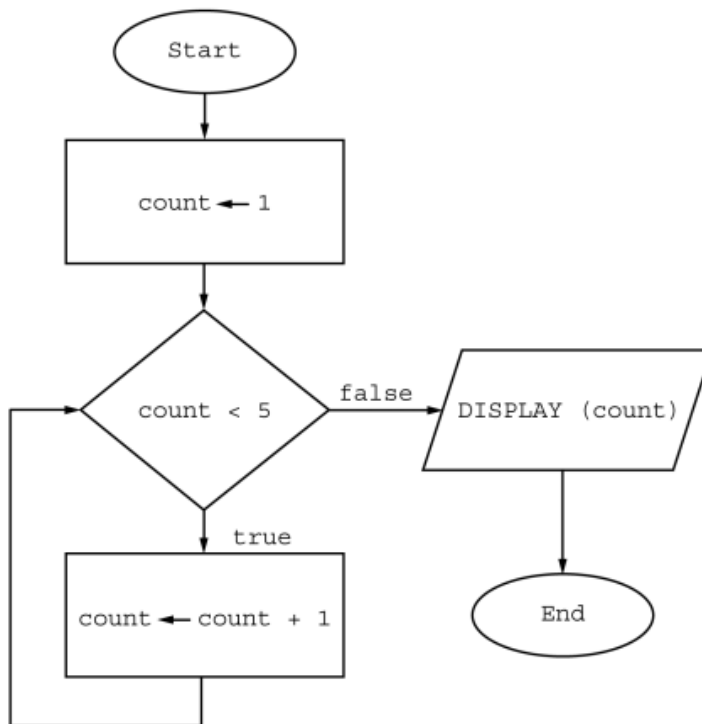
Which of the following statements is equivalent to the algorithm in the flowchart?

- (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.

`(numberOfAbsences ≤ 5) AND (gradePointAverage > 3.5)`

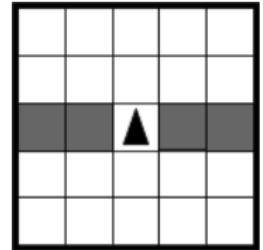
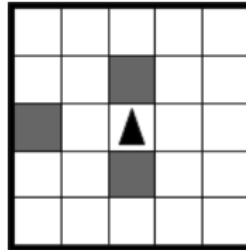
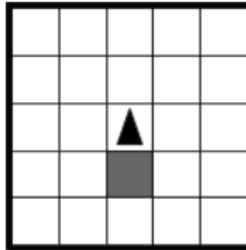
Draw a flowchart to represent the statement above. If the conditions above are met, the variable *academicAward* is true, otherwise it is false.

Name _____ Period _____

Skill 29.02 Exercises 1 thru 3

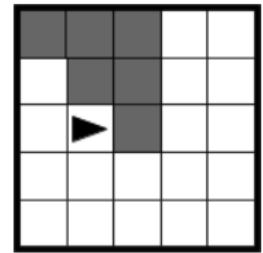
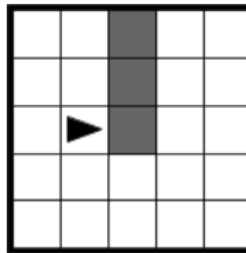
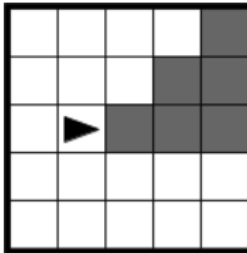
Basic If-Statements

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

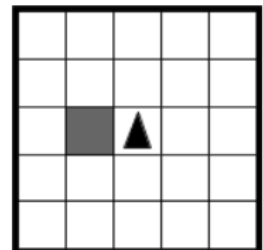
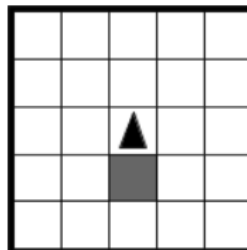
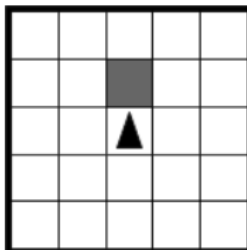


Sequential If-Statements

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



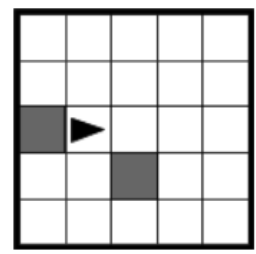
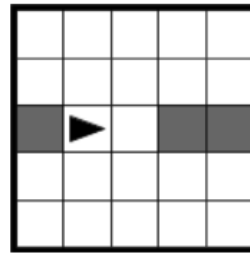
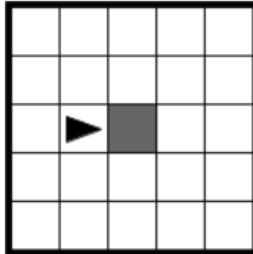
```
IF (CAN_MOVE ( left ))
{
    ROTATE_LEFT ()
    MOVE_FORWARD ()
}
IF (CAN_MOVE ( left ))
{
    ROTATE_LEFT ()
    MOVE_FORWARD ()
}
IF (CAN_MOVE ( left ))
{
    ROTATE_LEFT ()
    MOVE_FORWARD ()
}
}
```



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**

Declare a variable named sale. Assign the value true to it.

Now create an if statement. Provide the if statement a condition of sale. Inside the code block of the if statement, console.log() the string "Time to buy!".

Consider the block of code below,

- Re-write the code and add an if-statement to the code to check the age to see if the person is old enough to drive. (In most states you need to be 16 or older).
- Display a message if the person is old enough drive.

```
console.log("Driver Verification");
var age = prompt("Please enter your age");
console.log("It looks like you are old enough!");
```

Name _____ Period _____

Skill 29.04 Exercise 1

Consider the following students and their corresponding gpa's. Notice their rank is out of order! Write a program that puts the students in the correct order. The gpa and rank of each student can be accessed using the following syntax: `Bart.gpa`, `Bart.rank`

	gpa	rank
<code>var Bart</code>	3.5	1
<code>var Bugs</code>	3.8	3
<code>var Kyle</code>	3.1	2