|  |  |  |
| --- | --- | --- |
| Type | Instruction (Text) | Instruction (Block) |
| Basics | | |
| Variable Declaration | Variable ← Expression |  |
| Print | DISPLAY(Expression) |  |
| User input | INPUT( ) |  |
| Arithmetic Operators | * - \* / |  |
| Modulus | Variable1 MOD Variable2 |  |
| Random (Includes both ends) | RANDOM( a , b ) |  |
| Boolean Operators | ≠ Not equal to  = equal to  < Less than  > Greater than  ≥ Greater than or equal to  ≤ Less than or equal to |  |
| Compound Boolean Operators | Condition 1 **AND** Condition 2  Condition 1 **OR** Condition 2  **NOT** Condition |  |
| Conditionals | | |
| If | IF(Condition){ } |  |
| if / else | IF(condition){ } ELSE{ } |  |
| Looping | |  |
| Repeat (for loop) | REPEAT n TIMES { } |  |
| Repeat (While loop) | REPEAT UNTIL (Condition) { } |  |
| Lists/ Collections | |  |
| Declaration | list ← [value1, value2, value3] |  |
| Indexing | list [ i ] |  |
| Assigning value from one list to another | list [ i ] ← list [ j ] |  |
| Adding to a list | APPEND ( list, value ) |  |
| Inserting at an index (values shift) | INSERT ( list , i , value ) |  |
| Removing | REMOVE ( list, i ) |  |
| List length Method | LENGTH (list) |  |
| For Each Loop (lists only) | FOR EACH item IN list { } |  |
| Functions | |  |
| Defining a void function | PROCEDURE name (parameter1, parameter2 ……) { } |  |
| Defining a non-void function | PROCEDURE name (parameter1, parameter2 ……)  {  RETURN(Expression) } |  |
| Robot Code | | |
| Forward | MOVE\_FORWARD ( ) |  |
| Left | ROTATE\_LEFT ( ) |  |
| Right | ROTATE\_RIGHT ( ) |  |
| Check movement | CAN\_MOVE (direction) |  |
| Robot Grid Template (Delete cells to make a map) | | |
|  | | |

Questions



In which block will the Robot end the following code segment in?



1. 7
2. 8
3. 9
4. 10
5. 14



