Answers: Summative assessment

## Year 7 Programming

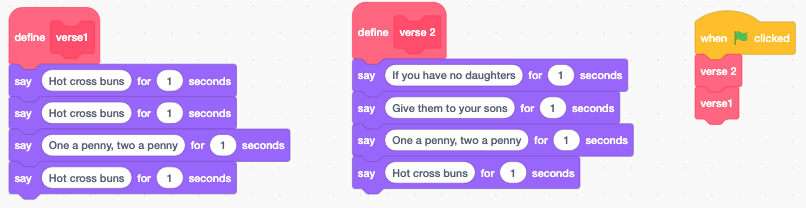
**Unit 1 Specific**

### Sequencing

Q1. Which one of the following segments of code would you find most appropriate to use in order to draw a blue square in Scratch?

|  |  |  |  |
| --- | --- | --- | --- |
| A |  | B |  |
|  | This is the **correct** answer as it provides all of the necessary and precise commands. |  | This missed the option of making the pen down and pen colour instructions. Learners might think this involves fewer lines of code and therefore is the correct answer. |
| C |  | D |  |
|  | There is no command to draw a square. The computer needs to be told specifically how to draw a square. |  | The commands ‘move forward’ and ‘turn right’ need additional information, such as how far and how many degrees. The computer will not guess this information or make assumptions like a human. |

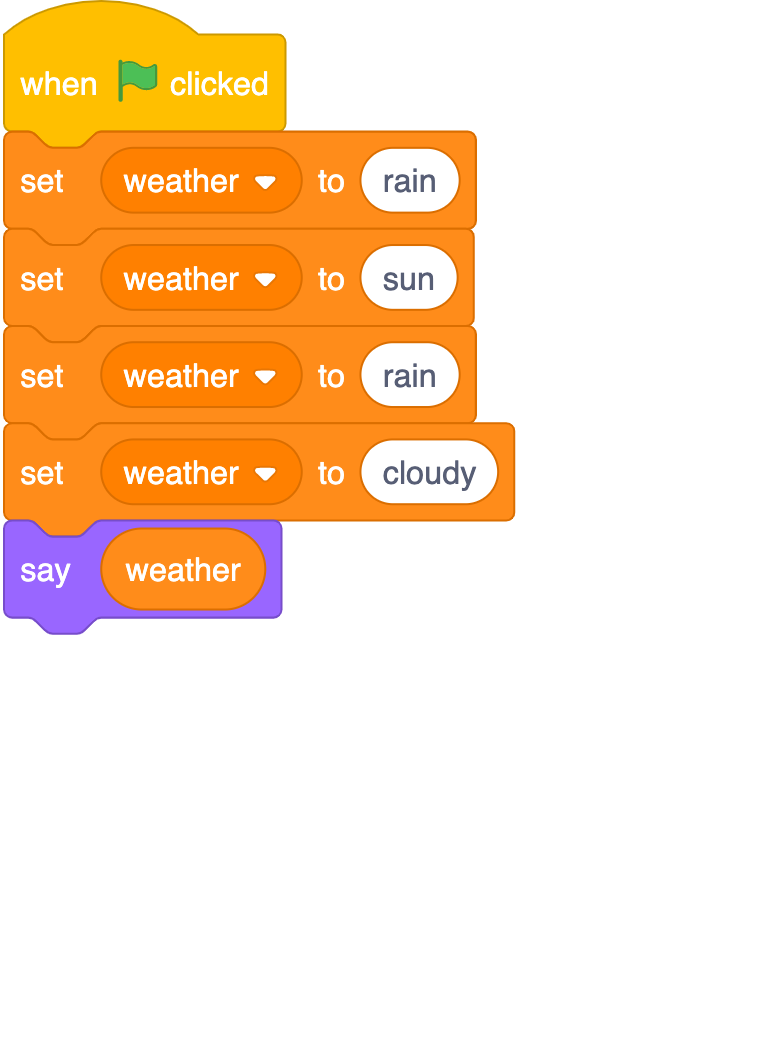
Q2. Would the following program correctly play the nursery rhyme *Hot Cross Buns*?



|  |  |
| --- | --- |
| **Answer** | **No** |
| Justification | This is to pick up on the misconception that subprograms are executed in the order in which they are defined. The program defines the subroutines in the correct order, but calls them in the incorrect order. |

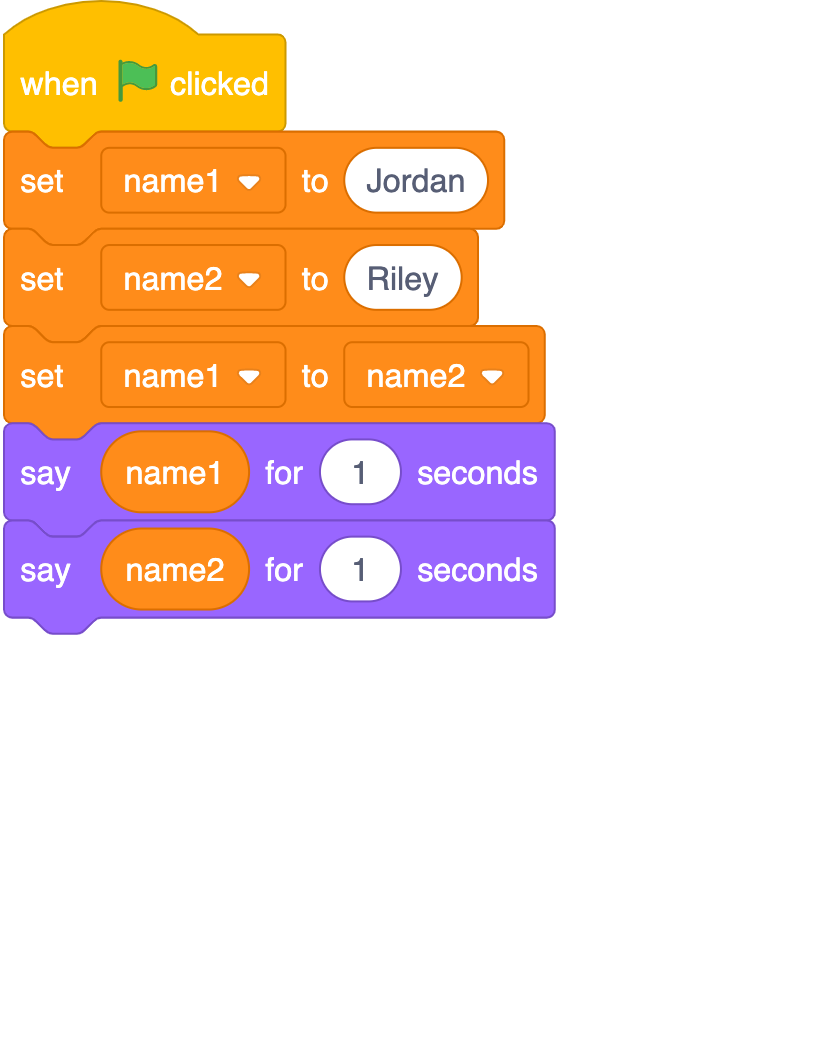
### Variables

Q3. What would the output of “say weather” be at the end of this program?



|  |  |  |
| --- | --- | --- |
| A | rain | *Incorrect: Misconception that the variable will only hold its initial value.* |
| **B** | **cloudy** | ***Correct****: Demonstrates that the learner understands that it doesn’t remember/store previous values once it has been overwritten.* |
| C | rain, sun, rain, cloudy | *Incorrect: Misconception that the variable stores a history of values.* |
| D | weather | *Incorrect: Misconception that the name of the variable is the value that is being held.* |

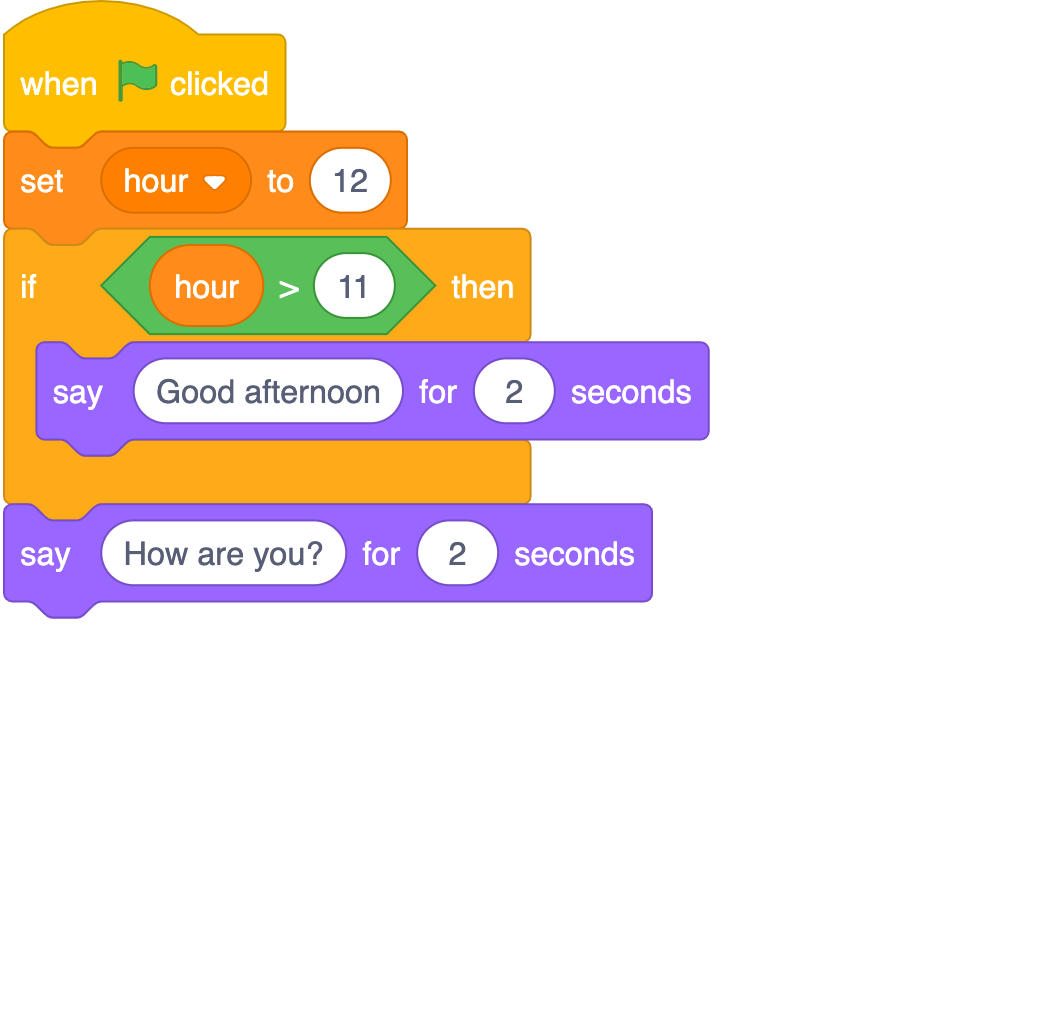
Q4. What value would be held by the ‘name1’ variable and the ‘name2’ variable at the end of this program?



|  |  |  |
| --- | --- | --- |
| A | name1 = Jordan  name2 = Riley | *Incorrect: ‘name1’ has been reassigned a new value on the third line.* |
| B | name1 = Riley  name2 = Jordan | *Incorrect: ‘name1’ has been reassigned to Riley (this is correct), but ‘name2’ has not changed from the original assignment, so Jordan is incorrect.* |
| C | name1 = Jordan  name2 = Jordan | *Incorrect: Assignment statements read from right to left and not the other way round.* |
| **D** | **name1 = Riley**  **name2 = Riley** | *This is the* ***correct*** *answer. Assignment statements are read from left to right so Riley will now also be held in ‘name1’.* |

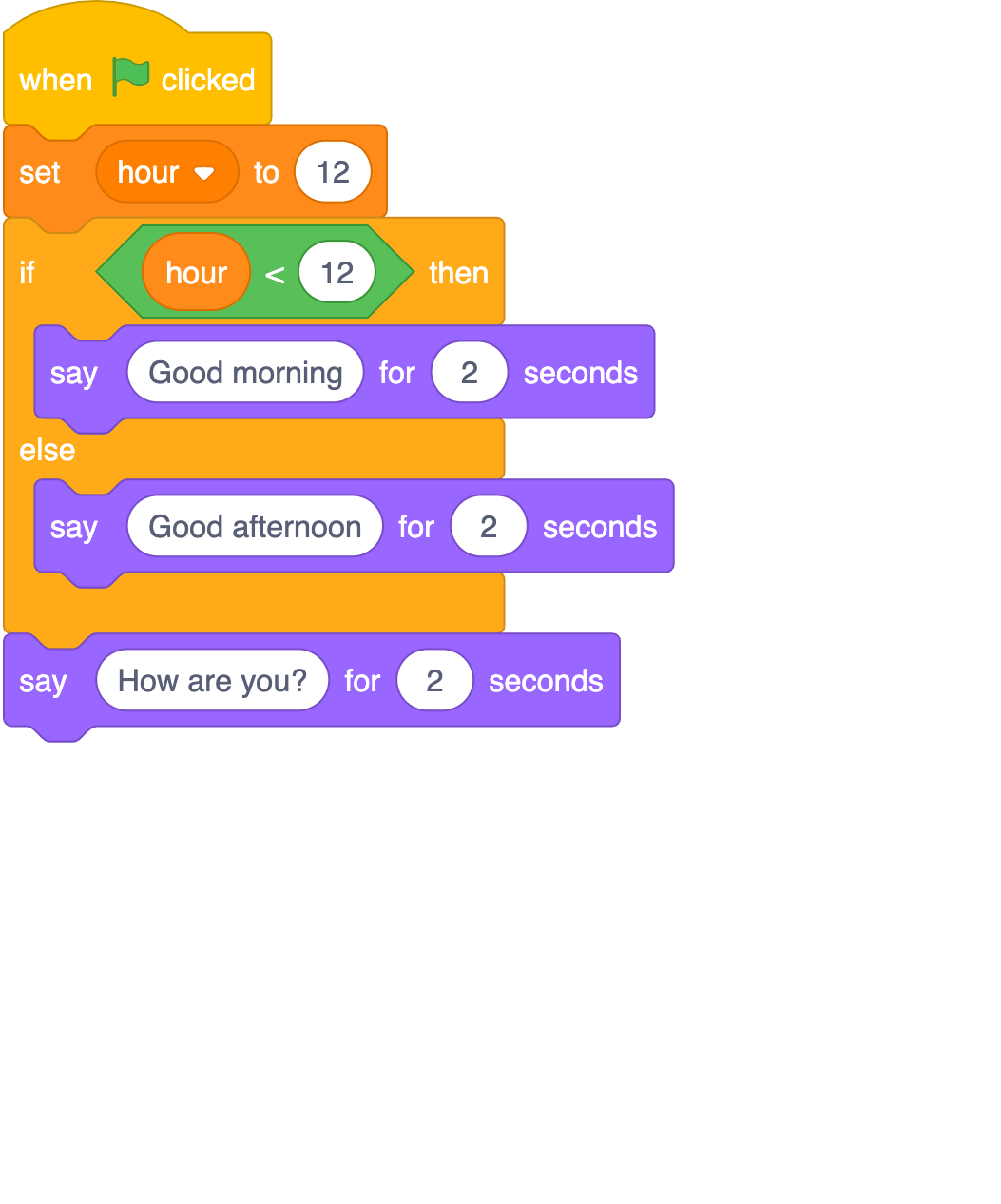
### Selection, conditions, and operators

Q5. What will be the output of this program when it is executed? (What will the sprite say)?



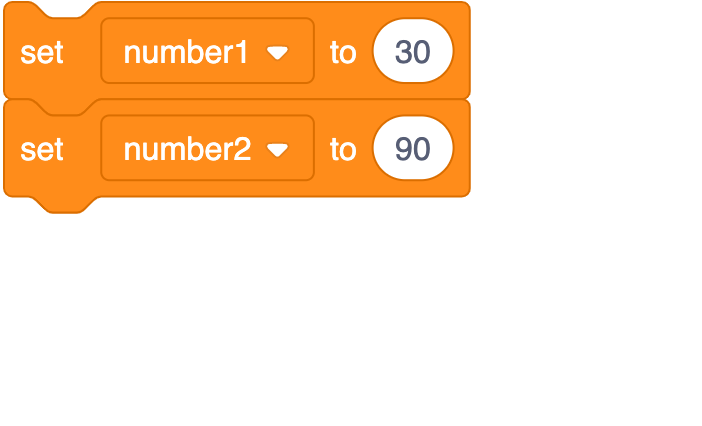
|  |  |  |
| --- | --- | --- |
| A | Good afternoon | *Incorrect: Although “Good afternoon” will be outputted, the final* ***Say*** *block is not the* ***Else****. “How are you?” will also be outputted.* |
| B | How are you? | *Incorrect: The condition will evaluate as ‘true’ and therefore “Good afternoon” will also be said.* |
| **C** | **Good afternoon**  **How are you?** | ***Correct****: The condition will evaluate as ‘true’ and therefore “Good afternoon” will be outputted. The “How are you?” will also be outputted as it is outside of the selection statement.* |
| D | Nothing will be outputted/said | *Incorrect: The condition will evaluate as ‘true’ and therefore “Good afternoon” will be outputted. The “How are you?” will also be outputted as it is outside of the selection statement and will be executed regardless.* |

Q6. What will be the output of this program when it is executed? (What will the sprite say)?



|  |  |  |
| --- | --- | --- |
| **A** | **Good afternoon**  **How are you?** | ***Correct:*** *Hour is equal to 12 and not less than 12. The condition evaluates as ‘false’, therefore the* ***Else*** *is executed as well as the* ***Say*** *block outside of the selection statement.* |
| B | How are you? | *Incorrect: This will be outputted, but where there is selection with* ***If*** *and* ***Else****, at least one will be executed.* |
| C | Good morning  Good afternoon  How are you? | *Incorrect: Misconception that the* ***Else*** *block is always executed.* |
| D | Good morning  How are you? | *Incorrect: Hour is equal to 12 and not less than 12. The condition evaluates as ‘false’. For “Good morning” to be outputted the statement condition would have had to have evaluated as ‘true’.* |

Q7. Which of the conditions would evaluate to ‘True’ with the following inputs?



|  |  |  |
| --- | --- | --- |
| A |  | *Incorrect: ‘number2’ is less than 90 but ‘number1’ is not greater than 30. As the* ***And*** *operator has been used, both conditions must be ‘true’ for the whole statement to evaluate as ‘true’.* |
| B |  | *Incorrect: Both conditions evaluate as ‘false’.* |
| C |  | ***Correct****: The* ***And*** *required both conditions to evaluate to ‘true’, which they do.* |
| D |  | ***Correct:*** *Both conditions evaluate as ‘true’. The* ***Or*** *operator requires one or the other. If both are ‘true’ the* ***Or*** *will still evaluate to ‘true’.* |

### 

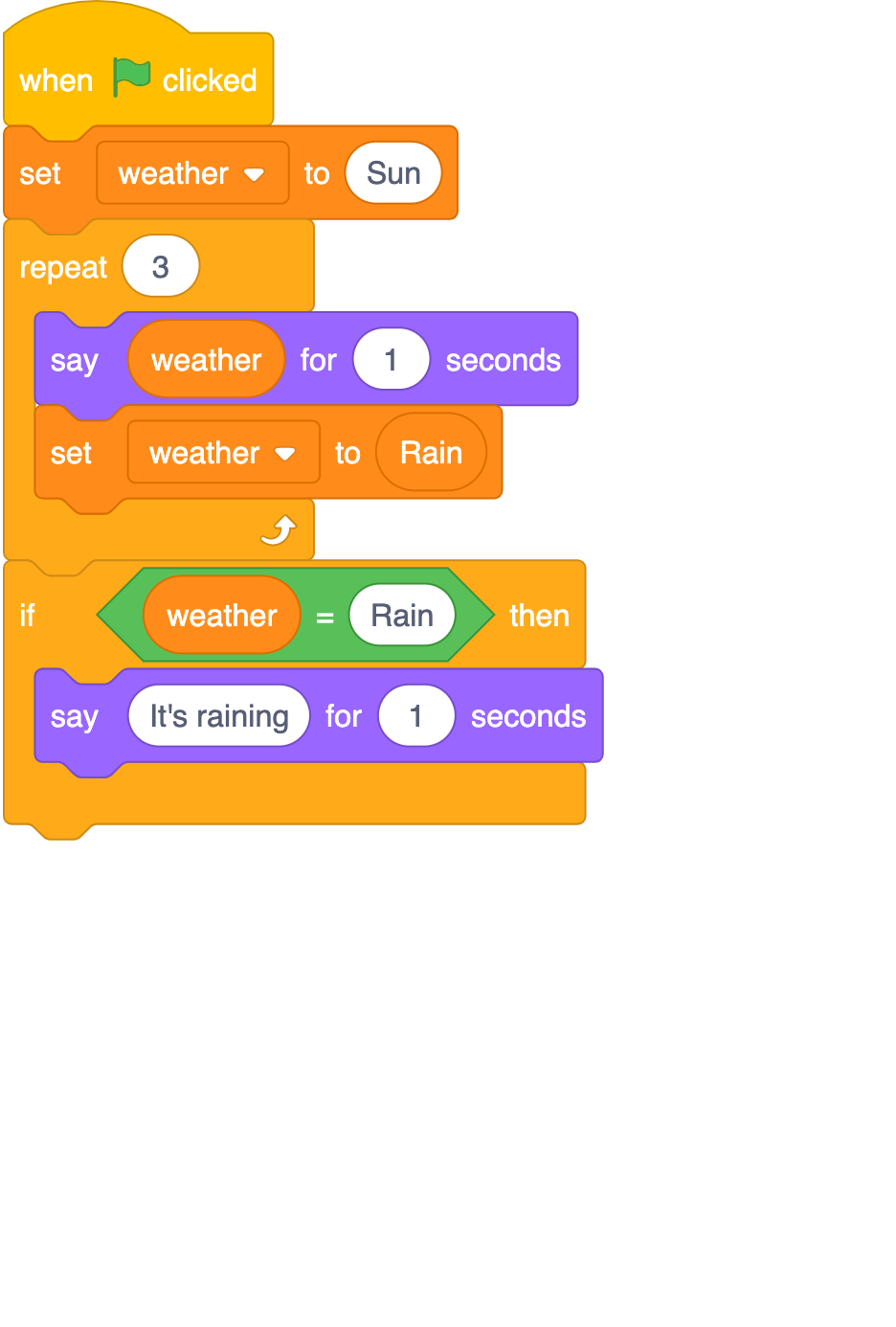
### 

### Count-controlled iteration

Q8.Which of the following blocks of code would make the Scratch Cat say “1, 2, 3” leaving a second in between each number?

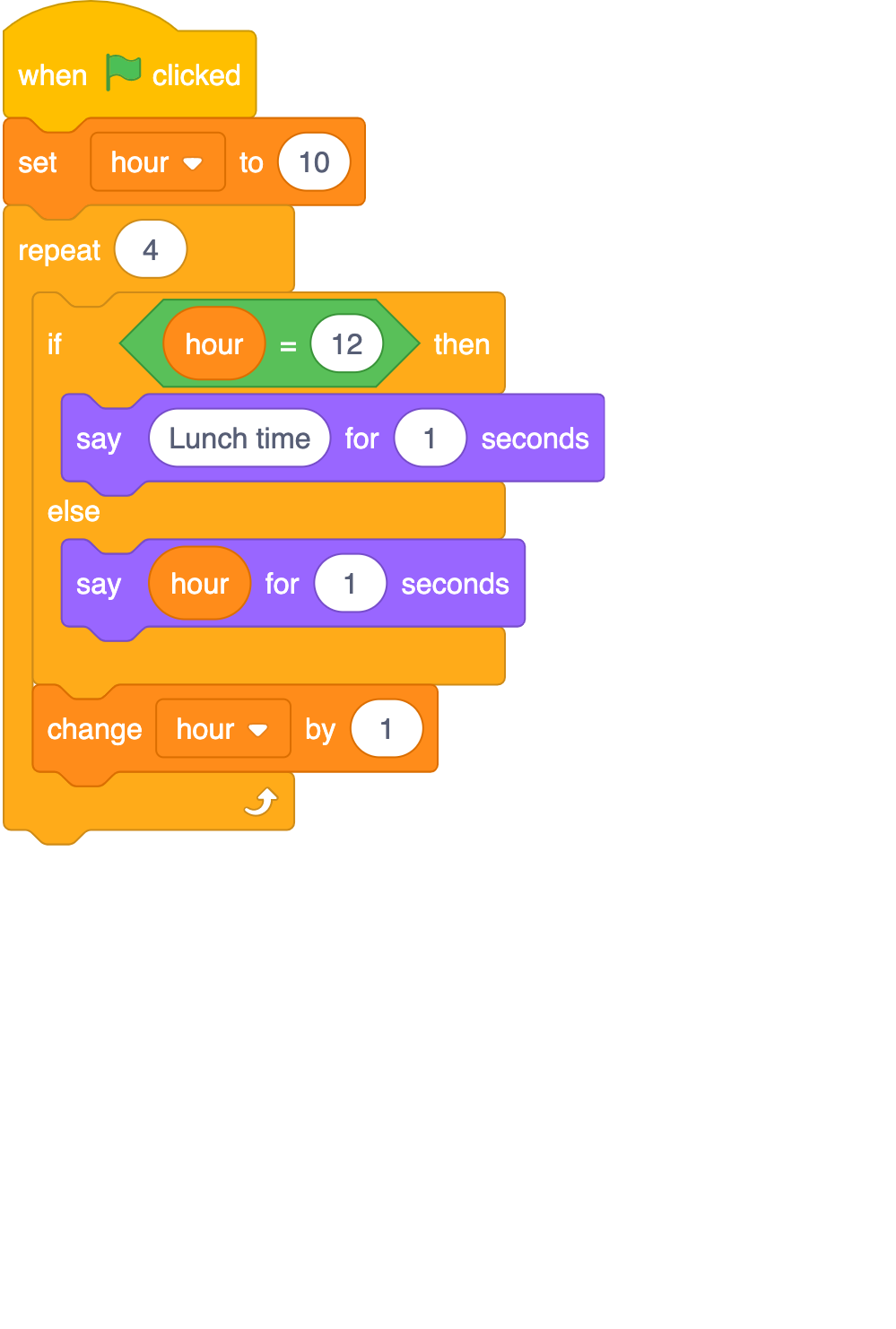
|  |  |  |  |
| --- | --- | --- | --- |
| A |  | B |  |
|  | Incorrect: The variable number is not incrementing and therefore this program will only output “1, 1, 1”. |  | Incorrect: The variable has been initialised to 1, but the value of the variable is changed before the number is outputted. This program will output “2, 3, 4”. |
| **C** |  | D |  |
|  | **Correct:** The variable has been initialised and incremented after the **Say** block. |  | Incorrect: The variable has not been given a value to start. Scratch will allow this, but by default will set the value of the variable to 0. This program will output “0, 1, 2”. |

Q9. What will be outputted (what will be said) when the following program is executed?



|  |  |  |
| --- | --- | --- |
| A | Sun, Rain, It’s raining | *Incorrect: Misconception — The loop does not break/stop iterating as soon as the condition of the selection statement is met.* |
| B | Sun, Rain, Rain | *Incorrect: The selection statement will be executed after the final iteration. The selection statement evaluates to ‘true’*  *and therefore it will output/say “It’s raining”.* |
| **C** | **Sun, Rain, Rain, It’s raining** | ***Correct****: All iterations will be completed before the selection statement evaluates the condition.* |
| D | Sun, It’s raining | *Incorrect: Misconception — The loop does not break/stop iterating as soon as the condition of the selection statement is met.* |

Q10. What will be the output of this program when it is executed? (What will the sprite say)?



|  |  |  |
| --- | --- | --- |
| A | 10, 11, Lunchtime, 12 | *Incorrect: This assumes that the number hasn’t incremented when the condition became ‘true’.* |
| **B** | **10, 11, Lunchtime, 13** | ***Correct:*** *The condition is ‘true’ when hour = 12 and therefore the program will say “Lunchtime”. On the other iterations the condition is being evaluated to ‘false’ and therefore the hour is outputted.* |
| C | 10, 11, 12, 13 | *Incorrect: This assumes that the condition never evaluates to ‘true’. It does when hour is equal to 12.* |
| D | 10, 11, Lunchtime, 12, 13 | *Incorrect: Misconception that the* ***Else*** *branch is always executed.* |

**Unit 2 Specific**

### Subroutines

Q11. Which of these sets of instructions is **not** a subroutine.

|  |  |  |  |
| --- | --- | --- | --- |
| **A** |  | B |  |
|  | **Correct: When green flag clicked** is not a subroutine, but it does use subroutine calls. |  | Incorrect: This is a subroutine. |
| C |  | D |  |
|  | Incorrect: This is a subroutine. |  | Incorrect: This is a subroutine. |

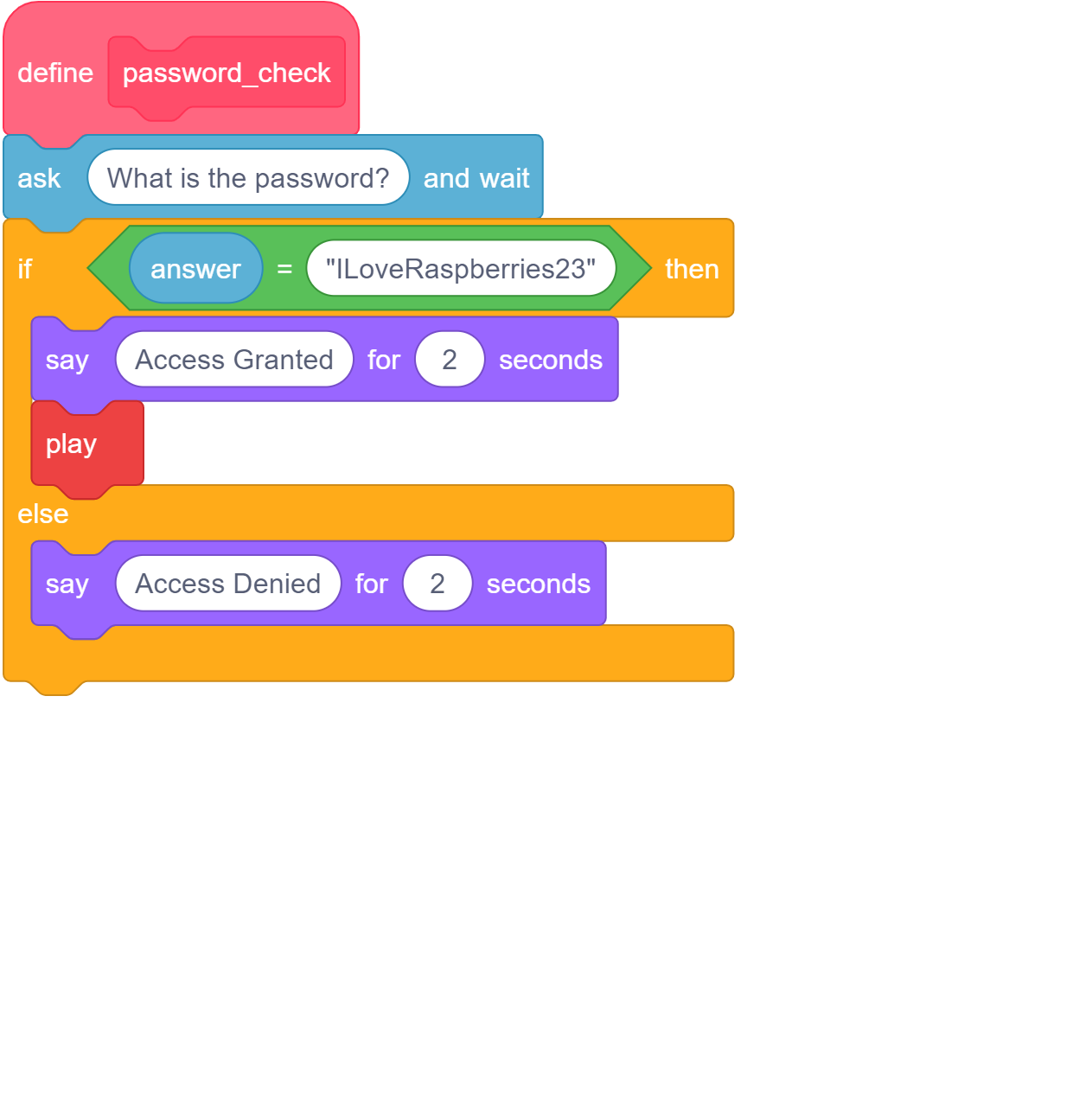
Q12. A large program has been broken down into subroutines to make the problem more manageable to solve. What is this process called?

Answer: **Decomposition**

This question is to check if learners are able to define the term decomposition.

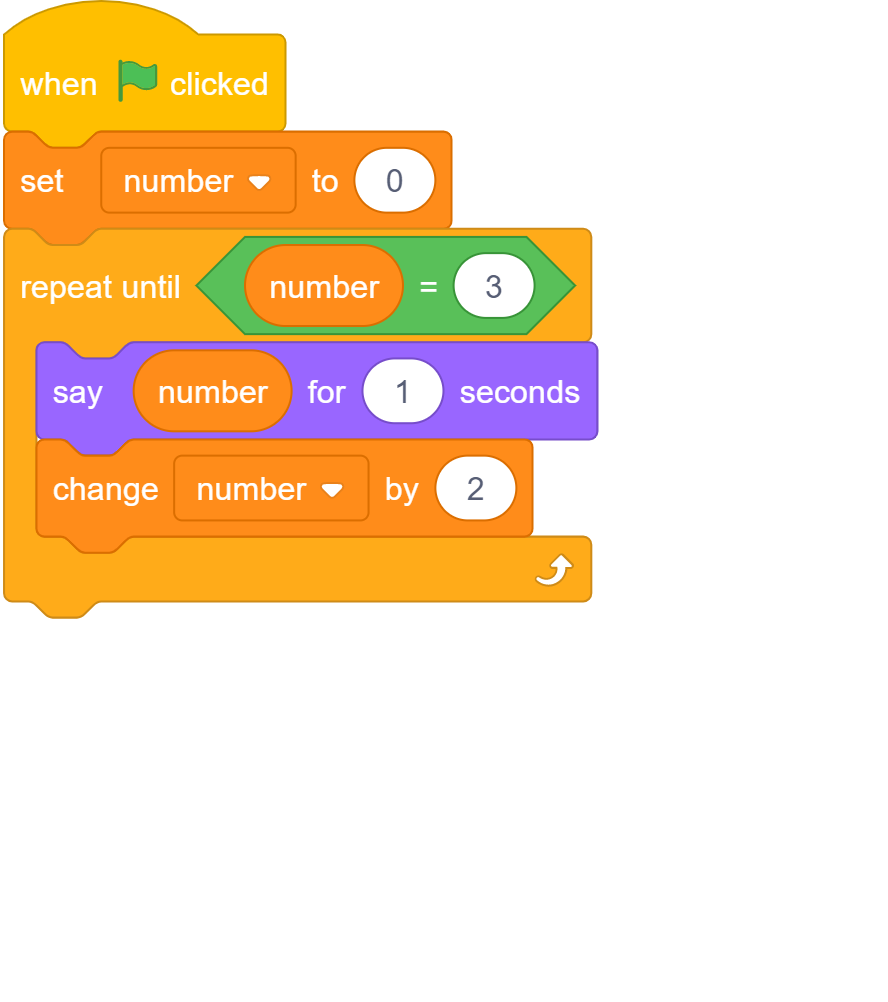
### Condition-controlled iteration

Q13. A learner has created the following piece of code to check if a password is correct or not. They want to improve it so that it will only give a maximum of three attempts before locking them out of the game for 60 seconds. What programming construct should they use for this?



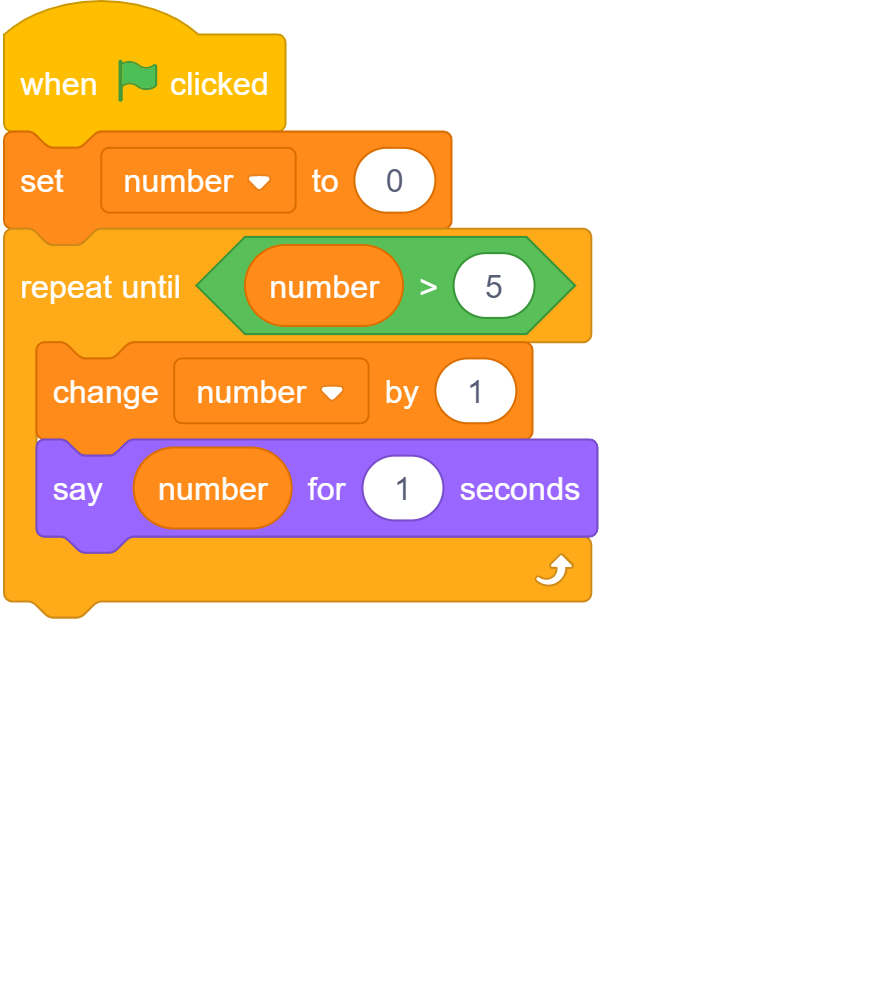
|  |  |  |
| --- | --- | --- |
| A | A **Forever** loop | *Incorrect: The code would get stuck in a loop forever and the user would keep getting asked for the password.* |
| **B** | A **Repeat until** loop | ***Correct****: This will allow the user access on the first., second, or third try but then lock them out.* |
| C | A **Repeat 3** loop | *Incorrect: This will ask the user the same question three times regardless of if they were correct or not.* |
| D | Three **If/else** statements | *Incorrect: This is inefficient and it will also ask the same question three times regardless of if they were correct or not.* |

Q14. A learner has created the following block of code, when will the loop terminate?



|  |  |  |
| --- | --- | --- |
| A | When ‘number’ holds the value ‘3’ | *Incorrect: The whole statement needs to be read here. The number variable will never hold the number ‘3’ so the loop will never end.* |
| B | When ‘number’ holds the value ‘4’ | *Incorrect: Learners might think that because the number is more than ‘3’, the loop will terminate, this is not the case.* |
| C | When ‘number’ is higher than ‘3’ | *Incorrect: Learners might think that this condition will be true when the value is higher than ‘3’ but the number needs to be exactly ‘3’ for this to be true.* |
| **D** | **Never** | ***Correct:*** *At each iteration the number is increased by ‘2’. This means that the number ‘3’ will be skipped and this condition will never be true. This is a common error that learners make when using* ***Repeat until*** *loops.* |

Q15. What will be the output of this program when it is executed? (What will the sprite say?)



|  |  |  |
| --- | --- | --- |
| A | Nothing | *Incorrect: Learners might confuse the more than symbol with the less than symbol and think that the loop will never run.* |
| B | 1, 2, 3, 4 | *Incorrect: Learners might think that the loop breaks as soon as number equals 5, this is not the case.* |
| C | 1, 2, 3, 4, 5 | *Incorrect: Learners might think that as soon as the number is more than 5, the loop will break. This is a common misconception. The loop will not end until all blocks of code in that loop have been run.* |
| **D** | **1, 2, 3, 4, 5, 6** | ***Correct****: The final* ***Say*** *block will say the number 6 before exiting the loop because the loop will go through all of the blocks of code within that loop before checking if the condition is true again.* |

### Evaluating the loop

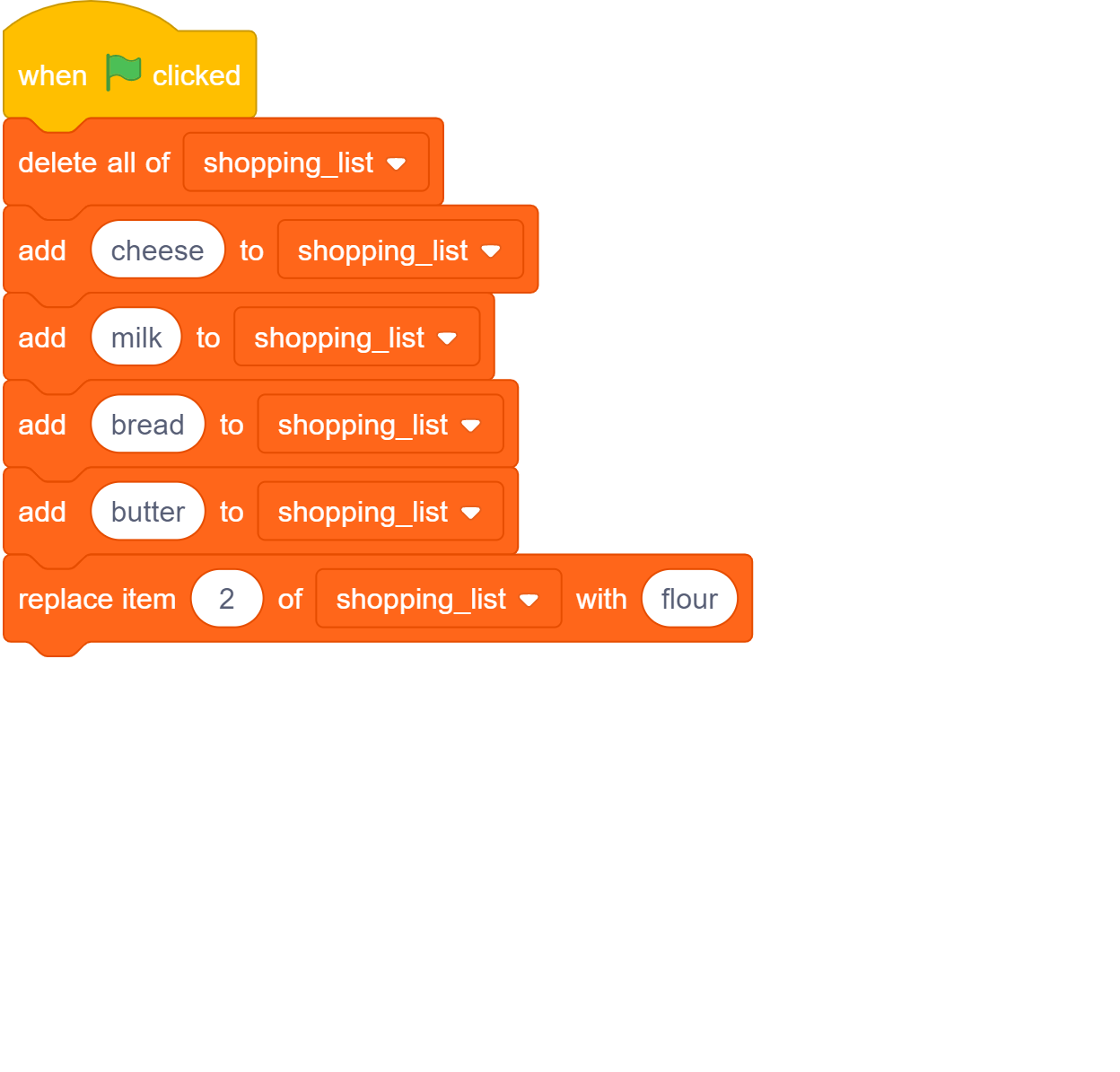
Q16. All of the blocks of code below will execute a countdown timer. Select the most appropriate block for this purpose when using Scratch.

|  |  |  |  |
| --- | --- | --- | --- |
| A |  | B |  |
|  | Incorrect: This block of code would take a lot of time to make for the programmer and is much harder to edit in future if the timings change. |  | Incorrect: This code block would work but it involves that ‘extra’ thought process by the programmer of creating a condition that isn’t needed. |
| **C** |  | D |  |
|  | **Correct:** This can be created by the programmer quickly and is also really easy to adjust if the timings change later on. |  | Incorrect: This block contains lots of unnecessary bits of code and is very inefficient to create. A **Stop ‘this script’** should be avoided if there is an easy work around. |

### 

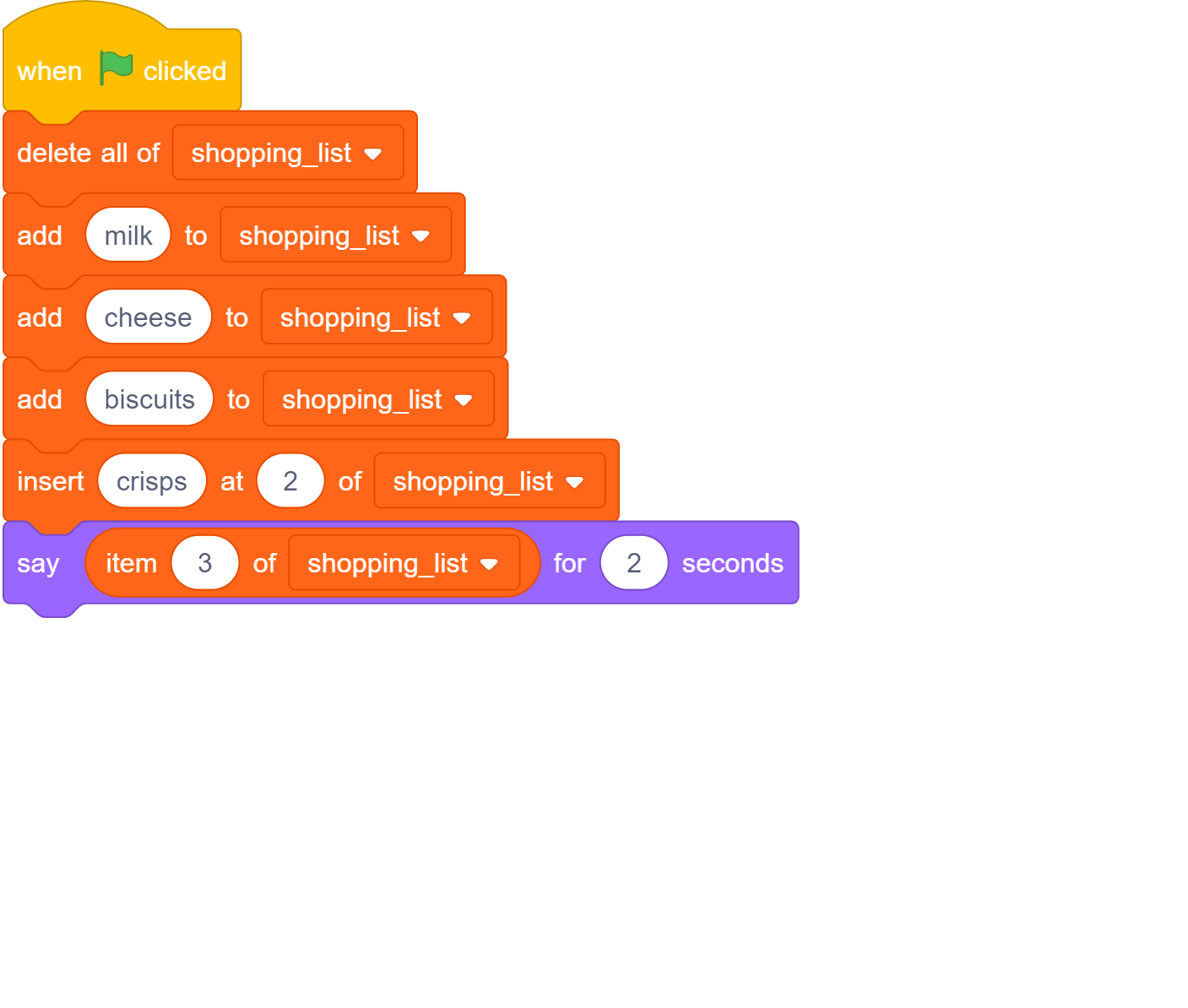
### Lists

Q17. The following block of code was used to create a shopping list and then replace a list item in ‘shopping\_list’. Which item was replaced with ‘flour’?



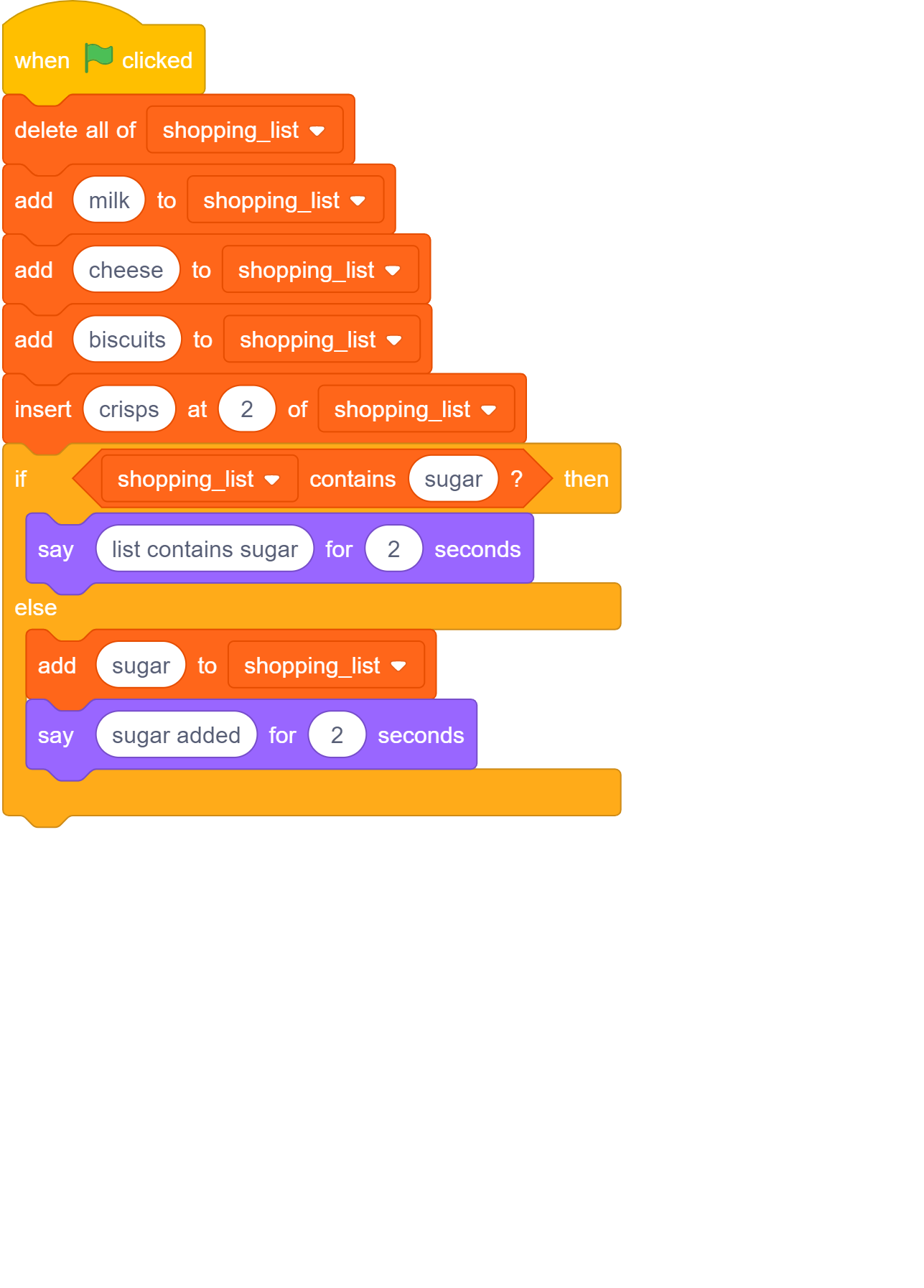
|  |  |
| --- | --- |
| **Answer** | **Justification** |
| **Milk** | This question is used to test if learners have remembered how indexing works in lists. |

Q18. What will be the output of this program when it is executed? (What will the sprite say?)



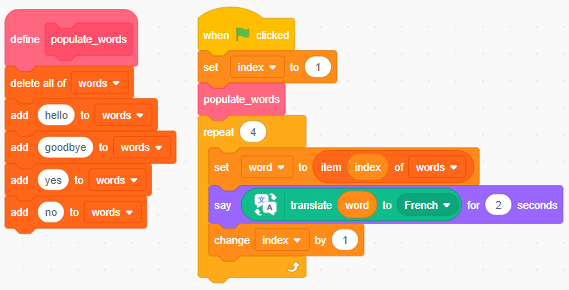
|  |  |
| --- | --- |
| **Answer** | **Justification** |
| **cheese** | This question is used to test if learners can trace the list items and also predict what the **Insert** block will do. |

Q19. What will be the output of this program when it is executed? (What will the sprite say?)



|  |  |
| --- | --- |
| **Answer** | **Justification** |
| **Sugar added** | This question is used to test if learners can determine what is already held in the list. It also tests learners’ understanding of the condition ‘list contains item’. |

Q20. A learner is using the **Translate** block to translate a list of words. What will be the output of this program when it is executed? (What will the sprite say?)



|  |  |  |
| --- | --- | --- |
| A | hello, goodbye, yes, no | *Incorrect: The* ***Translate*** *block has translated the word to French so the word will be said in French.* |
| B | hello, Hello, Hello, Hello | *Incorrect: The words will be said in French. Also,* ***Change index by 1*** *has been used which means that the word will increment through the list.* |
| **C** | **salut, au revoir, oui, non** | ***Correct:*** *The words will be translated to French because that block has been correctly applied. It will also increment through the list of words because the index is ‘changed by 1’ after each iteration.* |
| D | salut, salut, salut, salut | *Incorrect: The words will be said in French, which is correct. But,* ***Change index by 1*** *has been used, which means that the word will increment through the list.* |

Resources are updated regularly — the latest version is available at: [ncce.io/tcc](http://ncce.io/tcc).

This resource is licensed under the Open Government Licence, version 3. For more information on this licence, see [ncce.io/ogl](http://ncce.io/ogl).