Practical No: 3

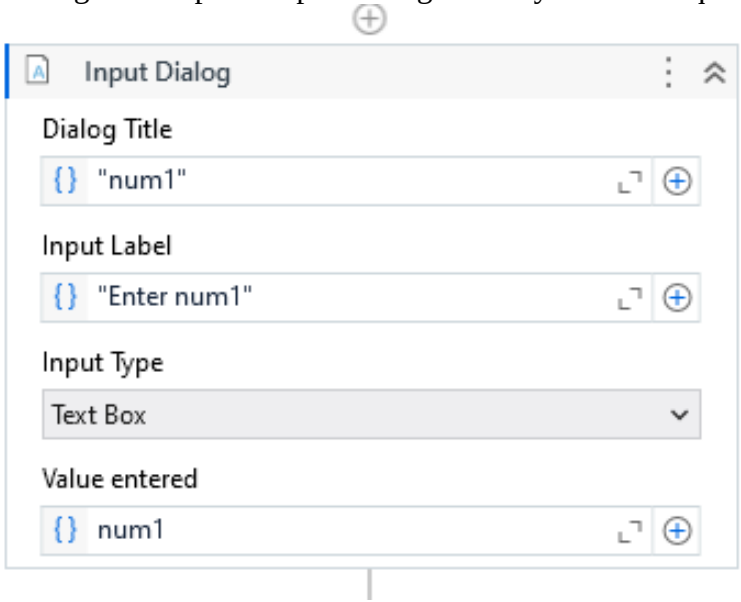
Decision Making and Looping statements.

AIM: A) Create an automation UiPath Project using Decision Making statements.

Steps with output

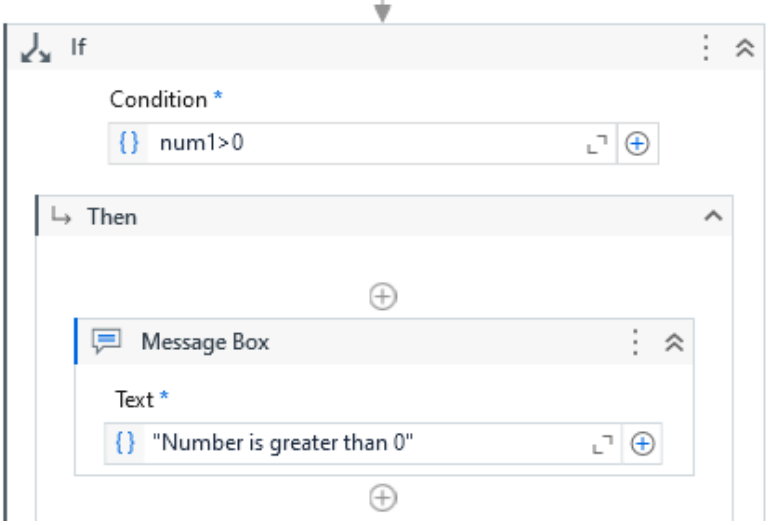
A. If Then

Step 1. Drag and drop an "Input Dialog" activity into the sequence.

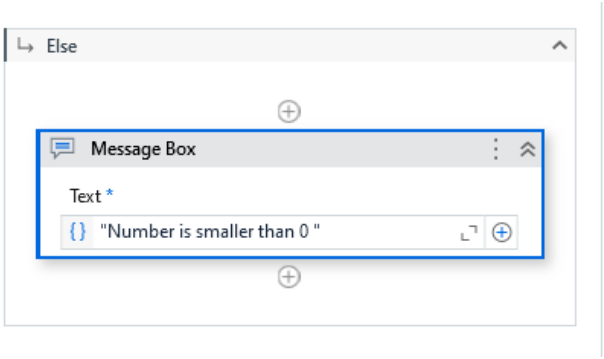


Step 2. Drag and drop an "If" activity below the "Input Dialog" activity.

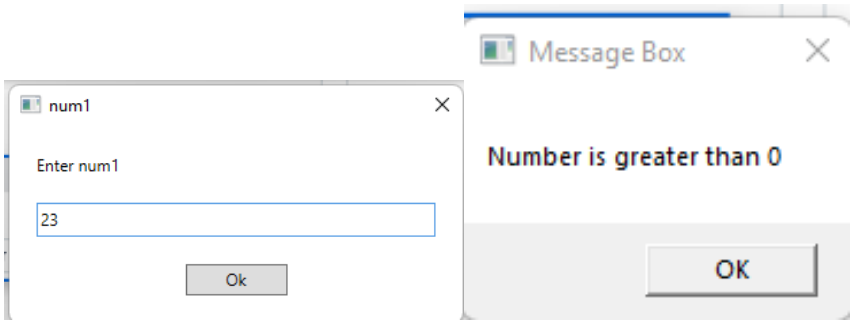
Step 3. Add a "Message Box" Activity (Then Branch):



Step 4. Add a "Message Box" Activity (Else Branch)

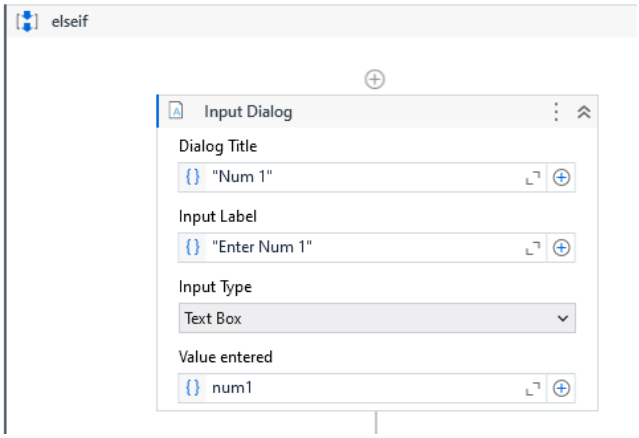


Output:

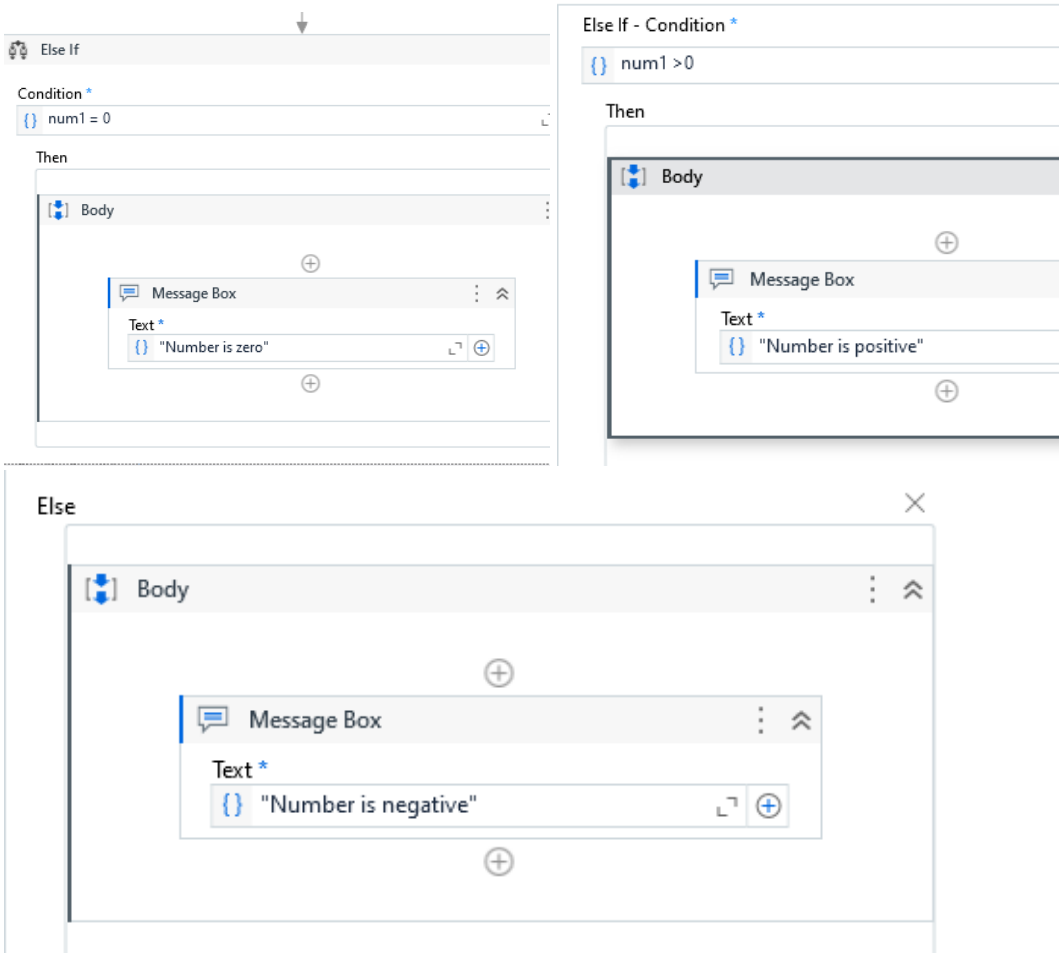


B. Else IF

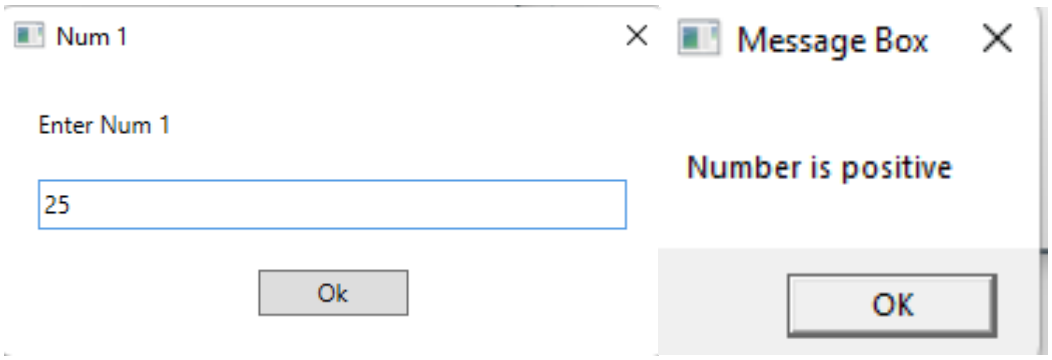
Step 1. Drag and drop an "Input Dialog" activity into the sequence**.**



Step 2. Add a "Message Box"

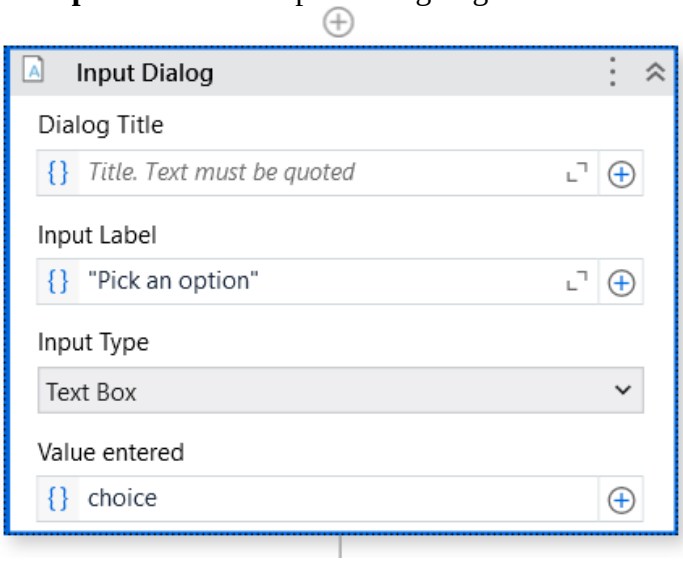


OUTPUT:

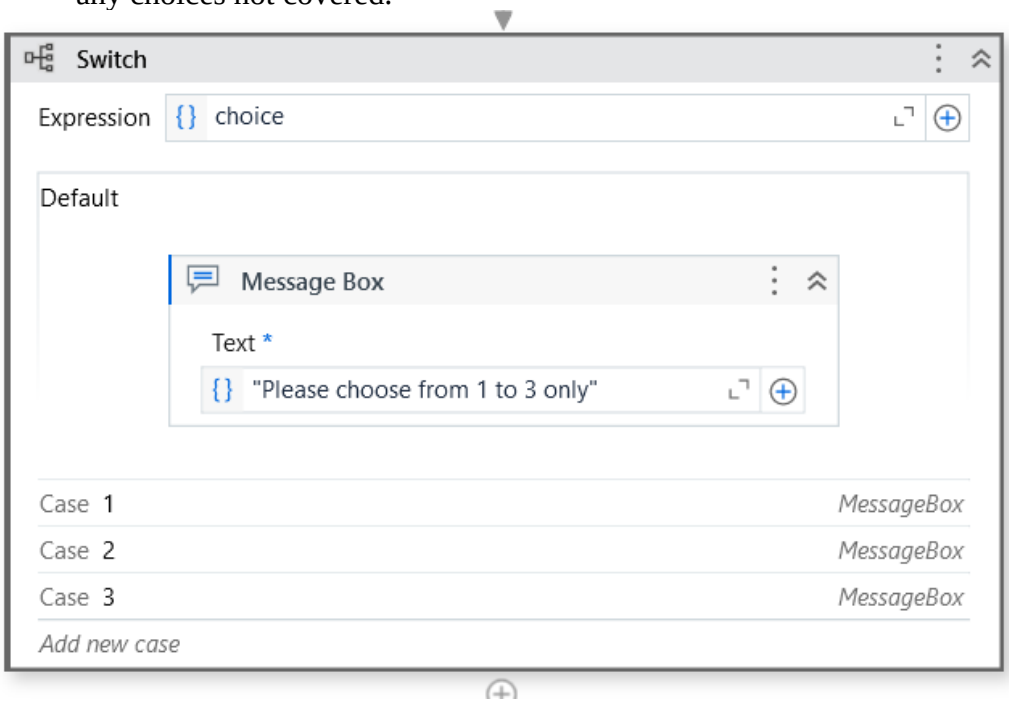


C. Switch

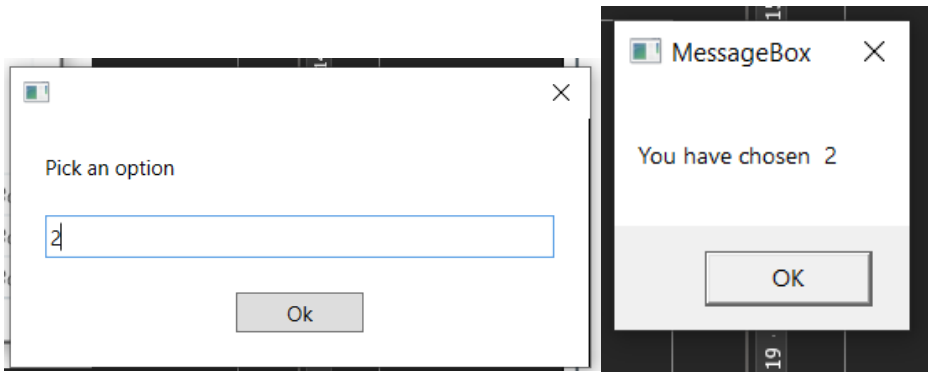
Step 1. Use an Input Dialog to get the user's choice.



Step 2. Set up a Switch Activity. Input the variable for expression. Add cases with associated message boxes for each choice. Don't forget to include a Default case for any choices not covered.



OUTPUT:



Learnings

A. If Then:

By using an Input Dialog followed by an If activity, we learned to create a decision-making statement. If a certain condition (e.g., number > 0) is true, a MessageBox in the "Then" branch is executed; otherwise, the "Else" branch with another MessageBox is triggered.

B. Else If:

Employing an Input Dialog and a Message Box, we grasped the concept of an "Else If" statement. Depending on the condition, a specific message is displayed in the MessageBox, demonstrating an alternative path in the execution.

C. Switch:

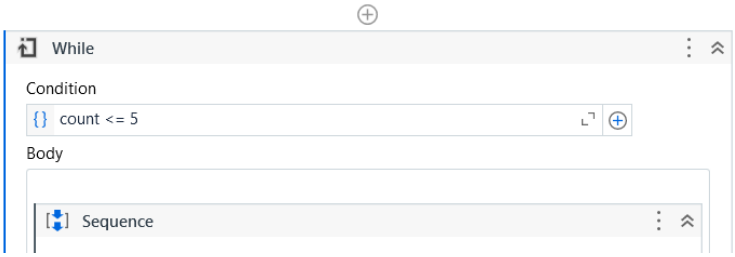
Through an Input Dialog capturing user choice and a Switch activity, we learned to handle multiple cases efficiently. The Switch statement directs the flow based on the user's choice, with each case associated with a specific action in a MessageBox.

AIM: B) Create an automation UiPath Project using looping statements

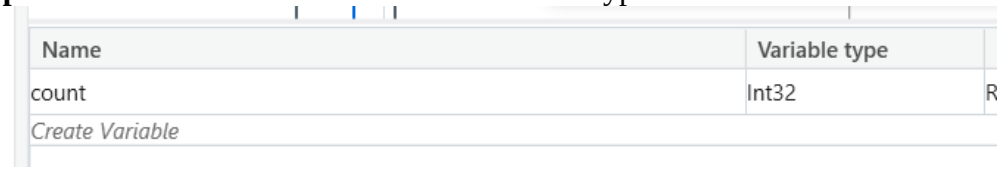
Steps with output

A. While Loop

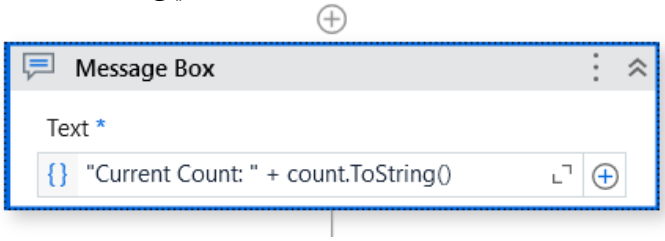
Step 1. Add while activity and set condition to -> count <=5



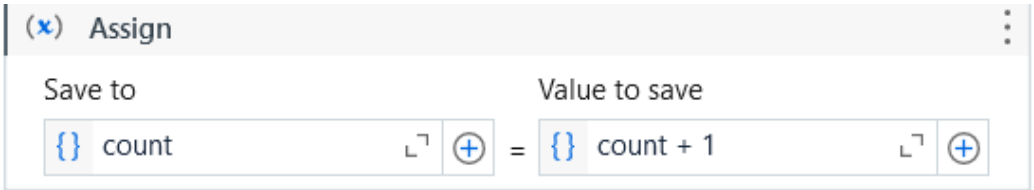
Step 2. Create variable count and set variable type to int32



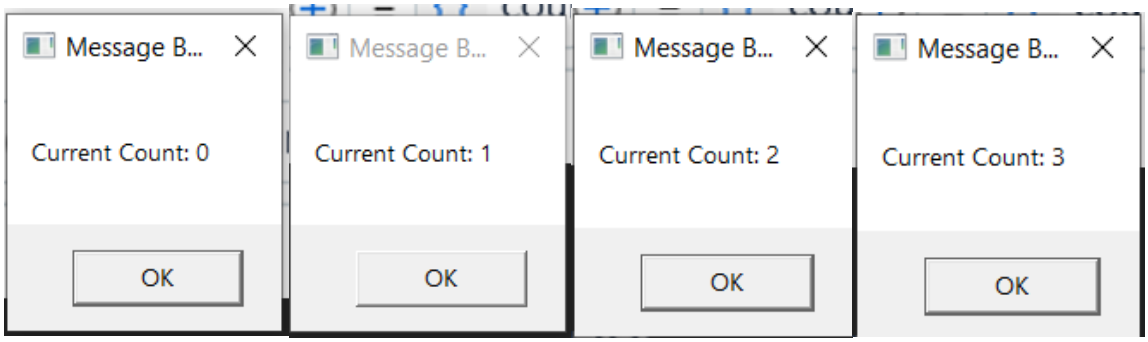
Step 3. Inside while body create message box and write message -> "Current Count: " + count.ToString()

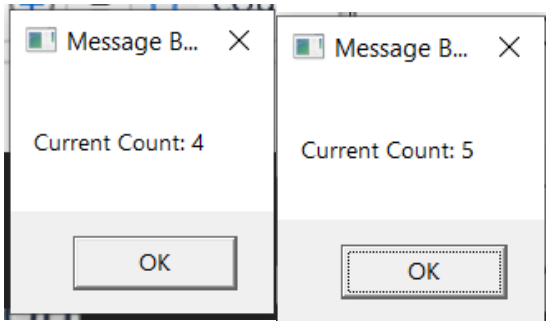


Step 4. Drag and drop assign activity inside while
Set the "To" field to Count.
Set the "Value" field to Count + 1.



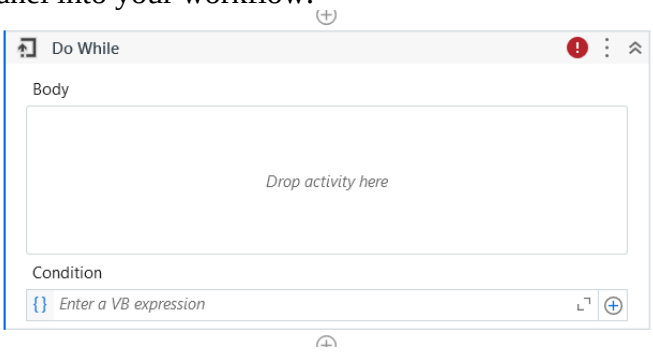
OUTPUT



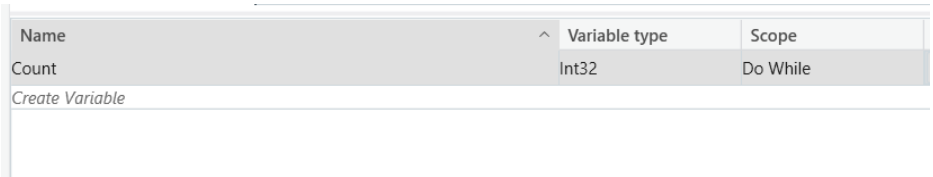


B. Do While

Step 1. Do While Loop: Drag and drop a "Do While" activity from the "Activities" panel into your workflow.

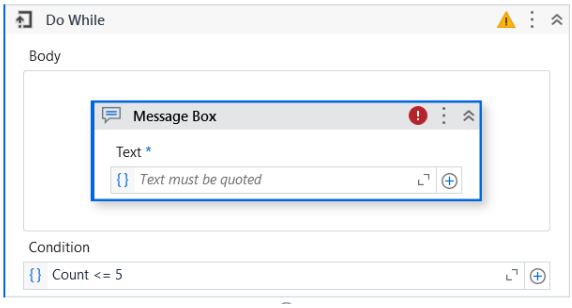


Step 2. Create a variable named ‘Count’ of type "Int32" to keep track of the current count.

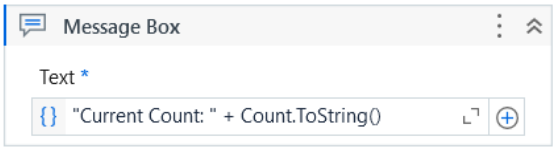


Step 3. we will use the condition: Count <= 5. This means the loop will continue as long as the Count variable is less than or equal to 5.

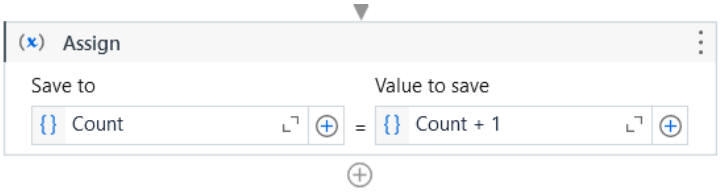
Step 4. Inside the "Do While" activity, drag and drop a "Message Box" activity.



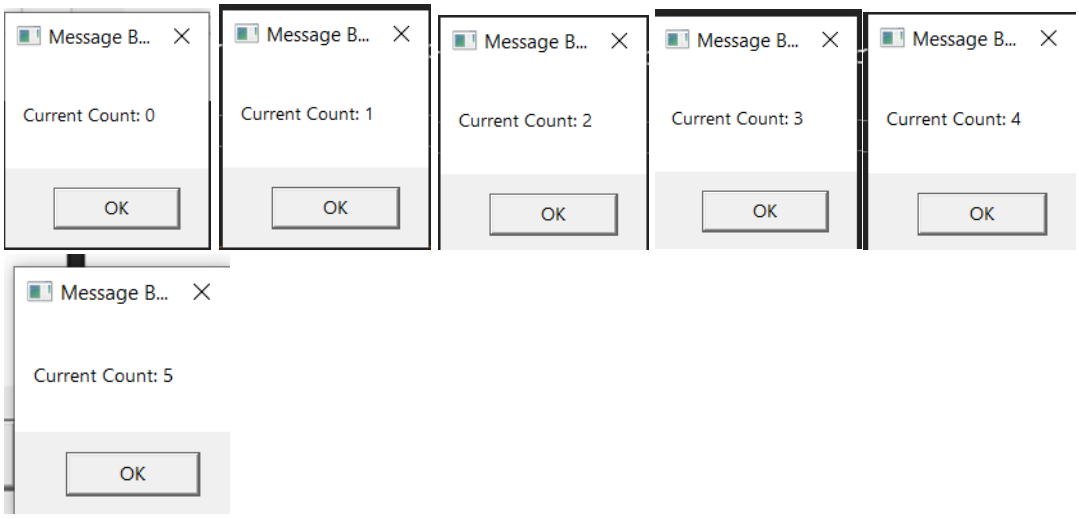
Step 5. Configure the Message Box Activity:



Step 6. Add an "Assign" activity inside the "Do While" loop:
Set the "To" field to Count.
Set the "Value" field to Count + 1.



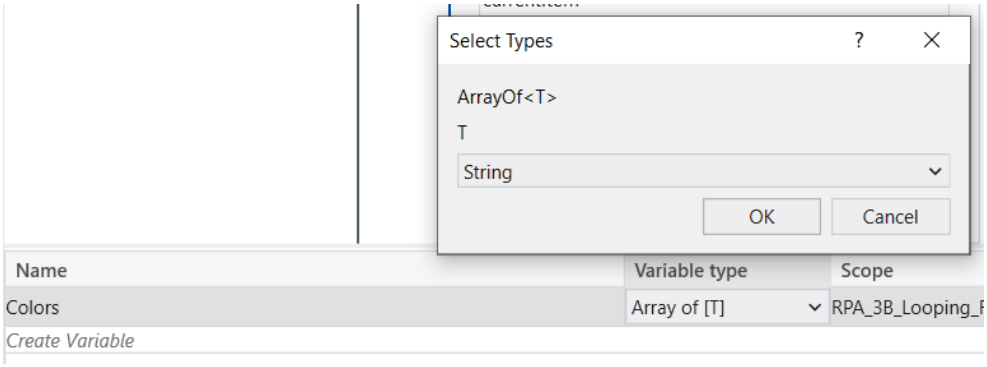
OUTPUT



C. FOR EACH

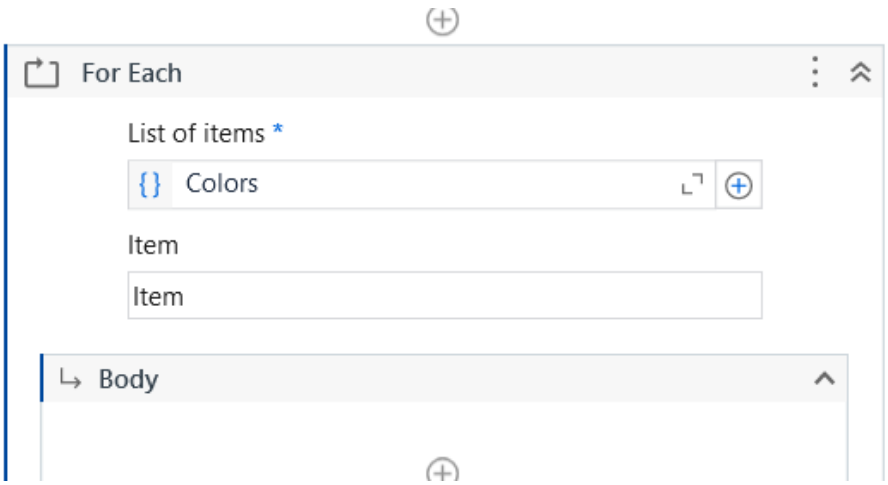
we'll create a "For Each" loop to iterate through a list of names and display each name using a message box.

Step 1. Add a List of Names:

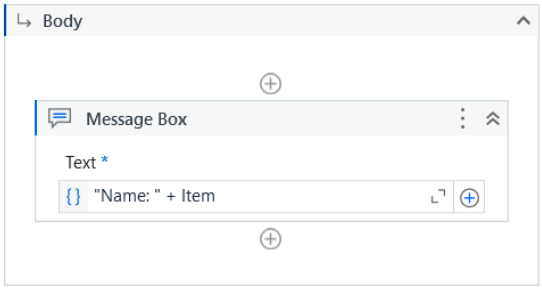


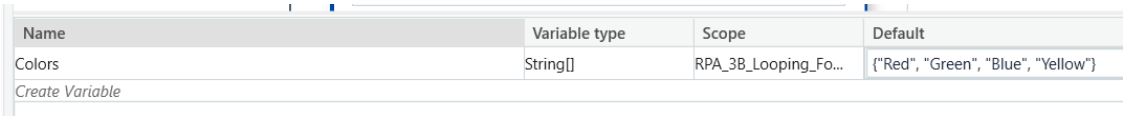
Step 2. In the "Default" value field of the variable, enter the list of colors enclosed in curly braces {} and separated by commas.

Step 3. Drag and drop a "For Each" activity from the "Activities" panel into your workflow.

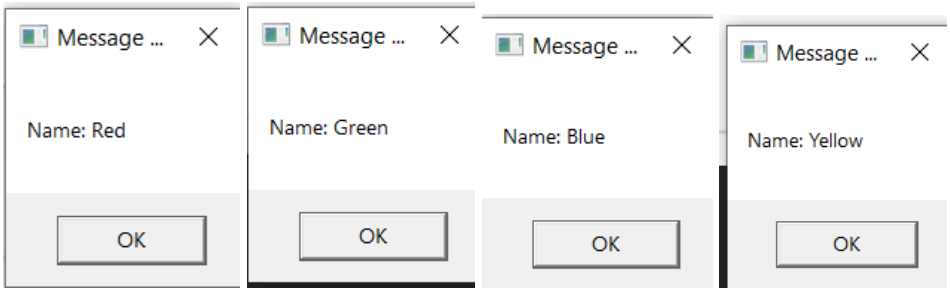


Step 4. Inside the "For Each" activity, drag and drop a "Message Box" activity.





OUTPUT:



Learnings

A. While Loop:

The While Loop iterates as long as the count is less than or equal to 5, displaying the current count in a Message Box and incrementing the count in each iteration.

B. Do While Loop:

The Do While Loop continues executing as long as the count is less than or equal to 5, showing the current count in a Message Box and incrementing the count within the loop.

C. For Each Loop:

The For Each Loop iterates through a list of names, displaying each name in a Message Box, showcasing the functionality of iterating through collections using a loop.