Week 2 – Lab/Practice Exercise Manual

2.5 Hello World

2.5.1 Objective

Build a workflow that prints “Hello World” in a message box.

2.5.2 Process Overview

- START
- Open UiPath Studio
- Add a Sequence activity
- Add a Message Box activity
- Enter the text “Hello World”
- STOP

2.5.3 Step by Step Process

Step 1: Open UiPath Studio.

Step 2: Create a new process and name it as “Hello World”

Step 3: Drag a Sequence activity from the Activities panel and drop it in the Designer panel.

Step 4: Name the Sequence activity as “Sequence – Hello World”

Step 5: Right click on the Sequence activity container and select *Annotations* from the context menu.

Step 6: Enter the annotation ‘This code will create a workflow to print “Hello World” in a message box.’

Step 7: Insert a Comment activity from the Activities panel within the Sequence activity.

Step 8: Enter the comment "'Message box' activity displays a message box with the specified text."

Step 9: Insert a Message Box activity below the Comment activity and name it as "Message Box – Hello World". Add the annotation "This activity displays data in a message box."

Step 10: In the text box of the Message Box activity, enter the text "Hello World".

Step 11: Save and run the workflow.

3.1 Variable Swapping

3.1.1 Objective

Build a workflow that swaps two numbers using a third variable.

- Ask the user to input two numeric values and store them in two variables.
- Swap values of both the variables with each other using a third variable.
- Display initial and swapped values of both the variables in the Output panel.

3.1.2 Process Overview

- START
- Use an Input Method activity to receive two numeric values from the user.
- Store the received values in two integer variables called **First_Input_Value**, and **Second_Input_Value**
- Declare a third integer variable called **Swapping_Support_Variable**
- Use Assign activity to assign the value of **First_Input_Value** to **Swapping_Support_Variable**
- Use second Assign activity to assign the value of **First_Input_Value** to **Second_Input_Value**
- Use third Assign activity to assign the value of **Second_Input_Value** to **Swapping_Support_Variable**
- Use Write Line activity to display initial and final values of **First_Input_Value** and **Second_Input_Value** in the Output panel.
- STOP

3.1.3 Step by Step Process

Step 1: Open UiPath Studio.

Step 2: Create a process and name it as “Variable Swapping”

Step 3: Drag a Sequence activity from the Activities panel and drop in the Designer panel.

Step 4: Name the Sequence activity as "Sequence – 'This code is for swapping two numbers using a third variable'"

Step 5: Insert a Comment activity from the Activities panel within the Sequence activity.

Step 6: Add comment "Taking input of two numbers from the user and swap them by using a third variable."

Step 7: Drag another Sequence activity from the Activities panel and insert below the Comment activity.

Step 8: Name the Sequence activity as "Sequence – 'For prompting the user to give input'".

Step 9: Right-click on the Sequence activity container and select *Annotations* from the context menu.

Step 10: Enter an annotation "This code is for swapping two numbers by using a third variable."

Step 11: Insert an Input Dialog activity within the second Sequence activity and name it as "Input – 'First Variable by User'".

Step 12: Right-click on the Input Dialog activity container and select Annotations from the context menu. Add annotation : "Taking User input and storing the value in "First_Input"".

Step 13: In the Input Dialog activity, enter values as shown below:

Title	Label
"First Value"	"Please enter the first numeric value: "

Step 14: In the Variables panel, create a variable for the above Input Dialog activity as shown below:

Name	Variable type	Scope	Default
First_Input_Value	Double	Sequence – 'This code is for swapping two	

		numbers by using a third variable'	

Step 15: Go to the Properties panel of the Input Dialog activity and insert **First_Input_Value** in its Output property.

Step 16: Insert the second Input Dialog activity below the previous Input Dialog activity, and name it as "Input – 'Second variable by User'".

Step 17: Right-click on the Input Dialog activity container and select *Annotations* from the context menu. Add annotation : "Taking User input and storing the value in "Second_Input_Value".

Step 18: In the second Input Dialog activity, enter values as shown below:

Title	Label
"Second Value"	"Please enter the second numeric value: "

Step 19: In the Variables panel, create a variable for the second Input Dialog activity as shown below:

Name	Variable type	Scope	Default
Second_Input_Value	Double	Sequence – 'This code is for swapping two numbers by using a third variable'	

Step 20: Go to the Properties panel of the Input Dialog activity and insert the variable **Second_Input_Value** in its Output property.

Step 21: Insert a Write Line activity from the Activities panel after the second Sequence activity, and name it as "Write Line – 'Value entered before swapping'".

Step 22: Right-click on the Input Dialog activity container and select *Annotations* from the context menu. Add annotation : "Enter the text to get the result in Output Panel".

Step 23: In the text box of the Write Line activity, enter the expression: **"First Value is:" + First_Input_Value.ToString + Environment.NewLine + "Second Value is:" + Second_Input_Value.ToString**

Step 24: Insert another Sequence activity from the Activities panel below the Write Line activity, name it as "Sequence – 'Swapping of numbers'" and annotate it as "This block of code will swap the values of the numbers entered".

Step 25: In the Variables panel, create a new variable as shown below:

Name	Variable type	Scope	Default
Swapping_Support_Variable	Double	Sequence – 'This code is for swapping two numbers by using a third variable'	

Step 26: Insert Assign activity in the third Sequence activity, name it as "Assign – 'Move the First_Input_Value to Swapping_Support_Variable'" and enter the annotation : "Swap Swapping_Support with First_Input_Value".

Step 27: In the Assign activity, enter values as shown below:

To	Value
Swapping_Support_Variable	First_Input_Value

Step 28: Insert second Assign activity below the previous Assign activity, name it as "Assign – 'Move the Second_Input_Value to First_Input_Value'" and Enter the annotation "Swap First_Input_Value with Second_Input_Value".

Step 29: In the second Assign activity, enter values as shown below:

To	Value
First_Input_Value	Second_Input_Value

Step 30: Insert third Assign activity below the second Assign activity, name it as "Assign – 'To swap Swapping_Support_Variable with Second_Input_Value'" and enter annotation: "Swap Second_Input_Value with Swapping_Support".

Step 31: In the second Assign activity, enter values as shown below:

To	Value
Second_Input_Value	Swapping_Support_Variable

Step 32: Insert Write Line activity below the third Sequence activity, name it as "Write Line – 'Swapped Result'" and enter annotation: "Enter the text to get the result in Output Panel".

Step 33: In the text box of Write Line activity, enter the expression: **"First Value after swapping is: " + First_Input_Value.ToString + Environment.NewLine + "Second Value after swapping is: " + Second_Input_Value.ToString"**

Step 34: Save and run the workflow.

4.1 Input Methods

4.1.1 Objective

Build a workflow that uses different Input Methods to input data in a Notepad.

- Open a Notepad file and type “Automation makes life easier”.
- Minimize the Notepad file using the ‘SimulateClick’ method.
- Type “Welcome to the new world of Automation” using the ‘SendWindowMessages’ method.
- Change the font type to Times New Roman, the font style to Italic, and increase the font size by 5.
- Close the Font window by clicking Enter.

4.1.2 Process Overview

- START
- Use Open Application activity to indicate a Notepad file.
- Use Type Into activity to enter “Automation makes life easier”.
- Minimize the Notepad window using the ‘Simulate Click’ method in Click activity
- Use Type Into activity to enter “Welcome to the new world of automation” using the ‘Send Window Messages’ method.
- Use Send Hotkey activity to send “Ctrl + A”.
- Use Click, Attach Window, and Type Into activities to change the font type to Times New Roman, font style to Italic and increase the font size by 5.
- Use Send Hotkey activity to close the Font window of the Notepad window.
- STOP

4.1.3 Step by Step Process

Step 1: Open a new Notepad file.

Step 2: Open UiPath Studio.

Step 3: Create a new process and name it as “Input Methods”.

Step 4: Drag and Drop a Sequence activity from the Activities panel into the Designer panel.

Step 5: Name the Sequence activity as “Sequence – ‘This is the code to test the input methods in UiPath’”

Step 6: Right-click on the Sequence activity container and select Annotations from the context menu. Add annotation "This block of code demonstrates a workflow that uses different Input Methods to input data in a Notepad."

Step 7: Drag and Drop an Open Application activity within the Sequence activity, rename it to “Open Application - Open Notepad” and Enter annotation: “Open application is used to open the blank notepad file.”.

Step 8: Click on the “Indicate element on screen” link and select the Notepad window.

Step 9: Click the hamburger button and select Edit Selector. In the bottom panel of the Selector Editor, rename the title of the notepad to ‘* - Notepad’. Click OK to save the changes.

Step 10: In the Do section of the Open Application activity, drag and drop a Type Into activity from the Activities panel, rename it to “Type First Text” and Enter annotation: “Types the first text in notepad file.”.

Step 11: Click on the “Indicate element on screen” link of the Type Into activity and select the editor area of the Notepad.

Step 12: In the text box of the Type Into activity, enter the text “Automation makes life easier”.

Step 13: Drag and drop a Click activity from the Activities panel below the Type Into activity, and rename it to “Click Minimize - Simulate Click”.

Step 14: Right-click on the Click activity container and select *Annotations* from the context menu.

Step 15: Add annotation “Minimizes the notepad window using SimulateClick.”

Step 16: Click on the “Indicate element on screen” of the Click activity, and select the Minimize button of the Notepad.

Step 17: In the Properties panel of the Click activity, check the box SimulateClick property.

Step 18: Drag and Drop another Type Into activity below the Click activity, rename it to "Type Second Text- Send window Messages" and Enter annotation: "Types the second text in Notepad file."

Step 19: Click on the "Indicate element on screen" link of the Type Into activity and select the editor area of the Notepad file.

Step 20: In the Properties panel of the second Type Into activity, check the box of SendWindowMessages property.

Step 21: In the text area of the second Type Into activity, enter the text "Welcome to the new world of automation."

Step 22: Insert a Send Hotkey activity below the second Type Into activity, rename it to "Send Hotkey - Select all text" and enter annotation: "Ctrl+A selects all the text in the notepad file."

Step 23: Click on the "Indicate element on screen" link and select the Notepad editor area.

Step 24: Check the box below **Ctrl** option in the Send Hotkey activity. Choose "A" from the dropdown menu below *Key* option.

Step 25: Insert a Click activity from the Activities panel, rename it as "Click Format Button." and enter annotation: "Click Format button from the menu section of the notepad file."

Step 26: Click on the "Indicate element on screen" link, select the *Format* button.

Step 27: Insert Attach Window activity below Click activity and rename it to "Attach Window - Format Menu Window". Right-click on Attach Window activity and select *Annotations* from the context menu.

Step 28: Add an annotation "Performs action on the format menu window"

Step 29: Select the *Format* menu from the Notepad window by clicking on "Indicate Element on screen" link.

Step 30: Inside the Do container of Attach Window activity, insert a Click activity and name it as "Click Font". Enter the annotation "Click Font button from the menu section of Format."

Step 31: Select the *Font* button from the Format menu of Notepad window.

Step 32: Insert another Attach Window activity after previous Attach Window activity, and rename it to “Attach Window - Font window”. Add the annotation “Performs Action on Font window.”

Step 33: Select the *Font* window by using “Indicate element on screen” link of Attach Window activity.

Step 34: Insert a Type Into activity within the second Attach Window activity, name it as “Type into - Font type” and enter annotation: “Types "Times New Roman" as Font Type.”

Step 35: Select the input field of the Font names section in the Font window.

Step 36: Type “Times New Roman” in the text field of Type Into activity.

Step 37: Drag and drop another Type Into activity, name it as “Type into – Font Style”, Enter annotation: “Types "Italic" as Font Style.” and select the input field of Font style section in the Font window.

Step 38: Type “Italic” in the text field of the Type Into activity.

Step 39: Insert a Get Text activity below the Type Into activity, name it as “Get Text – Font Size” and Enter annotation: “Text gets stored in a variable named "FontSize"”.

Step 40: Select the input field of the Font Size section in the Font window.

Step 41: In the Properties section of Get Text activity, save the result into a variable named as **FontSize**.

Step 42: Insert a Type Into activity, name it as “Type Into – New Font Size”, Enter annotation: “Incrementing Font size with '5'.” and select the input field of Font Size from the Font window.

Step 43: In the text area of Type Into activity, enter the expression: **(cint(FontSize) + 5).ToString**.

Step 44: Insert Send hotkey activity below the Type Into activity, name it as “Send Hotkey - Enter” and enter annotation: “Close the Font Window of Notepad Window”.

Step 45: Select *enter* from the dropdown menu of Key option.

Step 46: Save and run the workflow.

4.2 Starting Browser

4.2.1 Objective

Build a workflow that opens a browser and then opens UiPath's website.

- Open a browser.
- Open the URL – www.uipath.com.
- Display "Success" in a message box.

4.2.2 Process Overview

- START
- Use Open Browser activity and enter the website URL – "www.uipath.com".
- Use Message Box activity and enter text "Success".
- STOP

4.2.3 Step by Step Process

Step 1: Open UiPath Studio.

Step 2: Create a new process and name it as "Starting Browser".

Step 3: Insert a Sequence activity from the Activities panel into the Designer panel.

Step 4: Name the Sequence activity as "Sequence – This code is to understand about Open Browser activity".

Step 5: Right-click on the Sequence activity container and select Annotations from the context menu. Add annotation "This block of code demonstrates a workflow that opens a browser and then opens UiPath's website."

Step 6: Insert Open Browser activity from the Activities panel within the Sequence activity, name it as "Open Browser – Opens www.uipath.com" and enter the annotation "Open browser and redirects to the URL "www.uipath.com"".

Step 7: In the text box of the Open Browser activity, enter www.uipath.com.

Step 8: Insert a Message Box activity within the Do container of the Open Browser activity, name it as “Message Box – Success" and enter annotation: “Prints "Success"”.

Step 9: In the text box of the Message Box activity, enter the text “Success”

Step 10: Save and run the workflow.

4.3 Web Recording

4.3.1 Objective

Build a workflow using Web Recorder in UiPath Studio to Sign in to UiPath's website.

- Ask the user for his email address and password.
- Open the login page of UiPath's Website.
- Sign in to the website using the user's credentials.

4.3.2 Process Overview

- START
- Use two Input Dialog activities and take the email address and password from the user.
- Store received input in two variables.
- Use Open Browser activity and open URL – "www.uipath.com".
- Use Web Recorder in UiPath Studio to:
  - Click *Try UiPath Free* button.
  - Click *Login* the link on the next page.
  - Type in the user's email address and password.
  - Click on the *Login* button.
- STOP

4.3.3 Step by Step Process

Step 1: Open UiPath Studio.

Step 2: Create a new process and name it as "Sign in to UiPath's Website".

Step 3: Drag a Sequence activity from the Activities panel and drop in the Designer panel.

Step 4: Name the Sequence activity as "Sequence – UiPath Login using Web Recorder".

Step 5: Right-click on the Sequence activity container and select *Annotations* from the context menu.

Step 6: Enter the annotation "This block of code takes the credentials from the user and signs in to UiPath's website using Web Recorder."

Step 7: Insert an Input Dialog activity within the Sequence activity, name it as "Input Dialog – Email Address" and add Annotation "Takes Email Address from User."

Step 8: In the Input Dialog activity, enter values as shown below:

Title	Label
"Email Address"	"Enter your Email Address"

Step 9: In the Variables panel, create a variable for the above Input Dialog activity as shown below:

Name	Variable type	Scope	Default
EmailAddress	String	Sequence – 'UiPath Login using Web Recorder'	

Step 10: Go to the Properties panel of the Input Dialog activity and insert **EmailAddress** in its Output property.

Step 11: Insert another Input Dialog activity below the previous Input Dialog activity, name it as "Input Dialog – Password" and enter annotation: "Takes Password from User".

Step 12: In the Input Dialog activity, enter values as shown below:

Title	Label
"Password"	"Enter your Password"

Step 13: In the Variables panel, create a variable for the above Input Dialog activity as shown below:

Name	Variable type	Scope	Default
Password	String	Sequence – 'UiPath Login using Web Recorder'	

Step 14: Go to the Properties panel of the Input Dialog activity and insert **Password** in its Output property.

Step 15: Insert Open Browser activity below the Input Dialog activity, name it as "Open Browser - Opens "www.uipath.com" webpage" and add Annotation: "Open UiPath webpage."

Step 16: In the text area of Open Browser activity, enter URL "www.uipath.com".

Step 17: Run the program at this stage to ensure that the browser opens the URL. Keep the website opened to follow further steps.

Step 18: In the Design ribbon of UiPath Studio, click on the Recording button, and select *Web* from the dropdown menu.

Step 19: Select Click option from the Web Recording toolbar, and indicate *Try UiPath Free* button on UiPath's Website. It opens another webpage.

Step 20: Again select Click option from the Web Recording toolbar, and indicate *Login* link. It opens another webpage containing Login form.

Step 21: Select Type option from the Web Recording toolbar, and indicate input area of the Email Address of the login form. Enter the variable **EmailAddress** into it.

Step 22: Again select Type option from the Web Recording toolbar, and indicate input area of Password of the login form. Enter the variable **Password** into it.

Step 23: Select Click option from the Web Recording toolbar, and indicate *Login* button of the login form.

Step 24: Click Save and Exit option from the Web Recording toolbar.

Step 25: Drag the Web container from outside of the Open Browser activity container, and drop into the Do container of the Open Browser activity.

Step 26: Save and run the workflow.

5.1 If Statement

5.1.1 Objective

Build a workflow using If statement, which asks a user whether he will get the second Marshmallow or not.

- Ask the user, “Do you want to eat your first Marshmallow now or after 5 minutes?”
- If the user answers “Now”, respond with “Oops! You will not get the second Marshmallow.”
- If the user answers “After 5 minutes”, respond with “Congrats! You will also get the second Marshmallow.”
- If the answer is other than “Now” or “After 5 minutes”, respond with “Invalid Input”.

5.1.2 Process Overview

- START
- Use an Input Dialog activity to ask the user “Do you want to eat your first Marshmallow now or after 5 minutes?”
- Store user response in a string variable.
- Use an If activity to check the user response
    - If the answer is “Now”, use a Message Box activity to display “Oops! You will not get the second Marshmallow.”
    - If the answer is “After 5 minutes”, use a Message Box activity to display “Congrats! You will also get the second Marshmallow.”
    - If the answer is other than “Now” or “After 5 minutes”, use a Message Box activity to display “Invalid Input”.
- STOP

5.1.3 Step by Step Process

Step 1: Open UiPath Studio.

Step 2: Create a new process and name it as “If Activity”.

Step 3: Drag a Sequence activity from the Activities panel and drop it in the Designer panel.

Step 4: Name the Sequence activity as "Sequence – 'Marshmallow Game'".

Step 5: Right-click on the Sequence activity container and select *Annotations* from the context menu.

Step 6: Enter the annotation "This code is to ask the user whether he wants a second Marshmallow."

Step 7: Insert an Input Dialog activity within the Sequence activity and name it as "Input Dialog – 'Question'". Enter the annotation "Question to User".

Step 8: In the Input Dialog activity, enter values as shown below:

Title	Label
"Question"	"Do you want to eat your first Marshmallow? Choose among the following options: " + Environment.NewLine + "1. Now" + Environment.NewLine + "2. After 5 minutes"

Step 9: In the Variables panel, create a variable for the above Input Dialog activity as shown below:

Name	Variable type	Scope	Default
UserInput	String	Sequence – 'Marshmallow Game'	

Step 10: Go to the Properties panel of the Input Dialog activity and insert **UserInput** in its Output property.

Step 11: Insert If activity below the Input Dialog activity and name it as "If - To check if the user input is 'Now'". Enter annotation: "This activity judges the User Input whether it is "Now", "After 5 minutes" or "Invalid"".

Step 12: In the condition input area of If activity, enter the expression: **UserInput = "Now"**.

Step 13: Insert a Message Box activity in the **Then** section of the If activity and name it as "Message Box - Failed". Enter annotation: "Prints Fail message".

Step 14: In the Message Box activity, enter the text "Oops! You will not get the second Marshmallow."

Step 15: Insert second If activity, name it as "If - To check User input is 'After 5 minutes'", add annotation "Check whether the input contains 'After 5 minutes' or Invalid input" within the **Else** section of the first If activity.

Step 16: In the condition input area of second If activity, enter the expression: **UserInput = "After 5 minutes"**.

Step 17: Insert a Message Box activity in the **Then** section of the second If activity and name it as "Message Box - Success". Add annotation: "Prints Success message".

Step 18: In the Message Box activity, enter the text "Congrats! You will get the second Marshmallow."

Step 19: Insert another Message Box activity in the **Else** section of the second If activity and name it as "Message Box – Invalid Input". Add annotation: "Prints Invalid Input message".

Step 20: In the Message Box activity, enter the text "Invalid Input"

Step 21: Save and run the workflow.

5.3 Do While Loop

5.3.1 Objective

Build a workflow for a 'Guessing Game' with the following conditions:

- Generate a random number and prompt the user to input a number.
- In case of a wrong input, a message is displayed to the user stating, 'Please enter a lesser/greater number'.
- The loop keeps on running until the input number equals the generated number.

5.3.2 Process Overview

- START
- Use Input Dialog activity within Do While activity to get guessed number from the user.
- For Do While activity, set the condition to check guessed number is not equal to the actual number.
- Use Message Box activity to display “You Guessed it correct” for the correct match.
- Use If activity within Do While loop to check if the guessed number is equal to the actual number.
    - If correct, use Message Box activity to display “You Guessed it correct” for the correct match.
    - Use another If activity within Else section to check if the guessed number is greater than the actual number.
        - If correct, use Message Box activity to display “Please try a smaller number”.
        - If incorrect, use Message Box activity to display “Please try a greater number”.
- STOP

5.3.3 Step by Step Process

Step 1: Open UiPath Studio.

Step 2: Create a new process and name it as “Do While Loop”.

Step 3: Drag a Sequence activity from the Activities panel and drop it in the Designer panel.

Step 4: Name the Sequence activity as “Sequence – ‘Guessing Game”.

Step 5: Right-click on the Sequence activity container and select *Annotations* from the context menu.

Step 6: Enter the annotation :

“This block of code demonstrates a workflow using Do While statement for creating a 'Guessing Game' with the following conditions:

1. Generate a random number and prompt the user to input a number.

2. In case of a wrong input, a message is displayed to the user stating, 'Please enter a lesser/greater number’.

3. The loop keeps on running until the input number equals the generated number.”

Step 7: Create variables using Variables panel as shown below:

Name	Variable type	Scope	Default
RandomNo	Int32	Sequence – Guessing Game	25
GuessedNo	Int32	Sequence – Guessing Game	

Step 8: Insert Do While activity within the Sequence activity, name it as “Do While - Guessed Number <> Random Number”, add annotation “The loop iterates until it reaches the given condition”.

Step 9: Set its condition to **GuessedNo<>RandomNo**

Step 10: Insert an Input Dialog activity within Do While activity, name it as “Input Dialog - Guessed Number”, add an annotation: “Take Guessed Number as User input” and enter values as shown below:

Title	Label

"Number"	"Guess a number"

Step 11: In the Properties panel of Input Dialog activity, enter **GuessedNo** in the Output property.

Step 12: Insert If activity below the Input Dialog activity, name it as "If - User input equals Random Number", add an annotation: "This activity checks whether the User input is equal to the Random Number or not" and enter condition **GuessedNo=RandomNo**

Step 13: In the Then section, insert a Message Box activity and name it as "Message Box - Correct Guess". Add annotation: "Prints Correct Guess message".

Step 14: Enter a text "You Guessed it correct".

Step 15: Insert another If activity, in the **Else** section of the first If activity, and enter condition **GuessedNo>RandomNo.** Name it as "If- Guessed number is greater or smaller than Random Number", add an annotation: "This activity checks whether the user input is greater or smaller than the Random number."

Step 16: In the Then section, insert a Message Box activity, name it as "Message Box - Try Smaller Number", add an annotation: "Prints Smaller Number message" and enter the text "Please try a smaller number".

Step 17: In the Else section, insert a Message Box activity, name it as "Message Box - Try Greater Number", add an annotation: "Prints Greater Number message" and enter the text "Please try a greater number".

Step 18: Save and run the workflow.

5.4 While Loop

5.4.1 Objective

Build a workflow using While loop that tells the user if the input is a prime number or not.

- Ask the user to input a number.
- Check if it is a prime number.
- If the input number is prime, then display "It is a prime number" in a message box.
- If the input number is not prime, then display "It is not a prime number" in a message box.

5.4.2 Process Overview

- START
- Use Input Dialog activity and ask for any number from the user and store in a variable called **Number**.
- Create two more variables **i** and **c** with Variable Type as **Int32** and Default value as **2** and **0** respectively in the variables panel.
- Use While activity and set the condition to **i<Number**.
- Use If activity within the While activity and set the condition to **Number mod i=0**.
- Use an Assign activity within **Then** section and increment value of **c** by **1**.
- Use Assign activity after/below the If activity, and increment value of **i** by **1**.
- Use another If activity after/below the While activity and enter condition **c>0**.
- Use a Message Box activity within Then section to display “It is not a prime number”.
- Use a Message Box activity within the Else section to display “It is a prime number”.
- STOP

5.4.3 Step by Step Process

Step 1: Open UiPath Studio.

Step 2: Create a new process and name it as “While Activity”.

Step 3: Drag a Sequence activity from the Activities panel and drop it in the Designer panel.

Step 4: Name the Sequence activity as "Sequence – 'This is the code to test whether the input is a prime number or not.'"

Step 5: Right-click on the Sequence activity container and select *Annotations* from the context menu.

Step 6: Enter the annotation : "This block of code demonstrates a workflow using While loop that tells the user if the input is a prime number or not."

Step 7: Insert an Input Dialog activity within the Sequence activity, name it as "Input Dialog – 'To take the input from user'" and add annotation "Take User input as a Number".

Step 8: In the Input Dialog activity, enter values as shown below:

Title	Label
"Number"	"Enter a number"

Step 9: In the Variables panel, create three variables as shown below:

Name	Variable type	Scope	Default
Number	Int32	Sequence – 'This is the code to test whether the input is a prime number or not.'	
I	Int32	Sequence – 'This is the code to test whether the input is a prime number or not.'	2

c	Int32	Sequence – 'This is the code to test whether the input is a prime number or not.'	0

Step 10: Go to the Properties panel of the Input Dialog activity and insert **Number** in its Output property.

Step 11: Insert a While activity below the Input Dialog activity and name it as "While – 'To check if the number is a prime number or not'".

Step 12: Right-click on the While activity container, and select *Annotations* from the context menu.

Step 13: Add annotation "This block of code will check whether the number is prime. If it is, it will increment the value of 'c'."

Step 14: Inside the While activity, enter the condition as **i< Number**

Step 15: In the **Body** section of the While activity drag and drop a Sequence activity.

Step 16: Rename Sequence activity to "Sequence – 'Check the number using 'If'".

Step 17: Right-click on the Sequence activity container and select *Annotations* from the context menu.

Step 18: Add annotation "In this sequence using 'If' activity, the 'Number' is divided by 'i' until i=Number."

Step 19: Insert an If activity inside the Sequence activity.

Step 20: Inside the If activity, enter the condition as **Number Mod i = 0**.

Step 21: Inside the **Then** section of If activity, insert an Assign activity, and enter values as shown below:

To	Value
c	c + 1

Step 22: Change the Assign activity name to "Assign – 'Increment the value of c'".

Step 23: Right-click on the Assign activity container and select *Annotations* from the context menu.

Step 24: Add annotation “Incrementing the value of ‘c’ when ‘Number’ is found to be a prime number.”

Step 25: Below the If activity, insert another Assign activity and rename it to “Assign- Incrementing the value of ‘i'.

Step 26: In the Assign activity, enter the values as shown below:

To	Value
i	i+1

Step 27: Right-click on the Assign activity container, and select *Annotations* from the context menu.

Step 28: Add annotations "Incrementing the value of 'i' whenever the loop iterates".

Step 29: Below the While activity, insert an If activity and name it as “If – Print the message”.

Step 30: Right-click on the If activity container, and select *Annotations* from the context menu.

Step 31: Add annotation “This block of code will print the message in a message box whether the input is Prime or not.”

Step 32: Inside the If activity, enter the condition **c>0**.

Step 33: In the **Then** section, insert a Message Box activity and name it as “Message Box - Not a prime number”. Add annotation “Displays that the number is not a prime.”

Step 34: Enter text “It is not a prime number.”

Step 35: In the **Else** section, insert another Message Box activity and name it as “Message Box - Is a prime number”. Add annotation “Displays that the number is not a prime.”

Step 36: Enter text “It is a prime number.”

Step 37: Save and run the workflow.