Practical No. 1

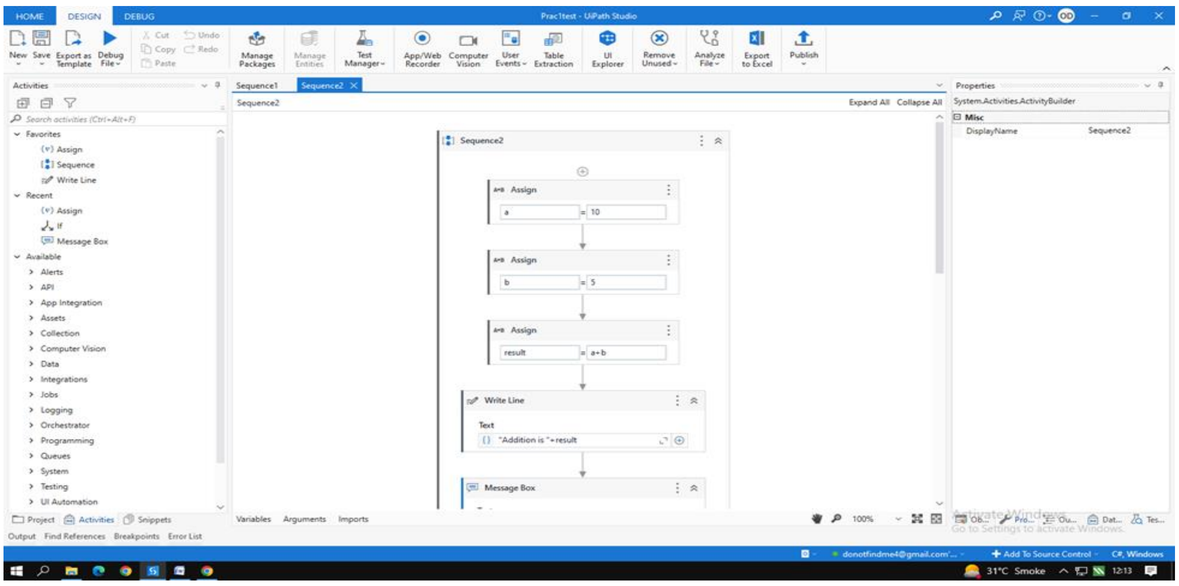
Aim : Automate UiPath Number Calculation (Subtraction, Multiplication, Division of numbers).

Steps:

1. Create a new sequence.
2. In the sequence, drag and drop an assign box to assign a variable. Assign a variable "a" with value 10.
3. On the variables tab, create the variable and assign the variable type as Int32.
4. Repeat steps 2 and 3 to create another variable "b" of integer type and give it the value 15.
5. Add a third assign box and create a variable "result" that will store the value of the operation between a and b. (a+b). This variable will also be integer.
6. Once done, drag and drop a write line box and add: "Addition is" + result since the output will be in string format.
7. Repeat steps from 5 and 6 thrice and change the operation as result=a-b, result=a*b, result=a/b.
8. Run the automation.

Output:

This will give the output of addition, subtraction, multiplication, division between variables a and b



Sequence2

Assign: a = 10

Assign: b = 5

Assign: result = a+b

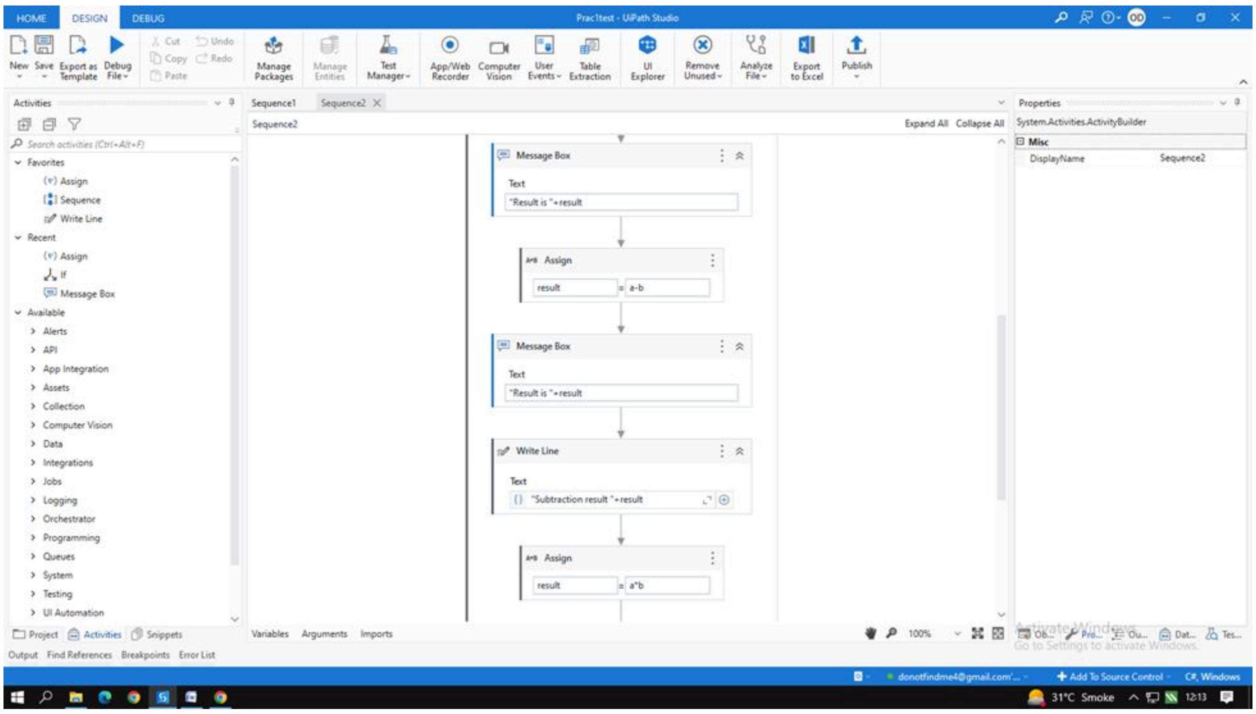
Write Line

Text: "Addition is "+result

Message Box

Properties: System.Activities.ActivityBuilder

DisplayName: Sequence2



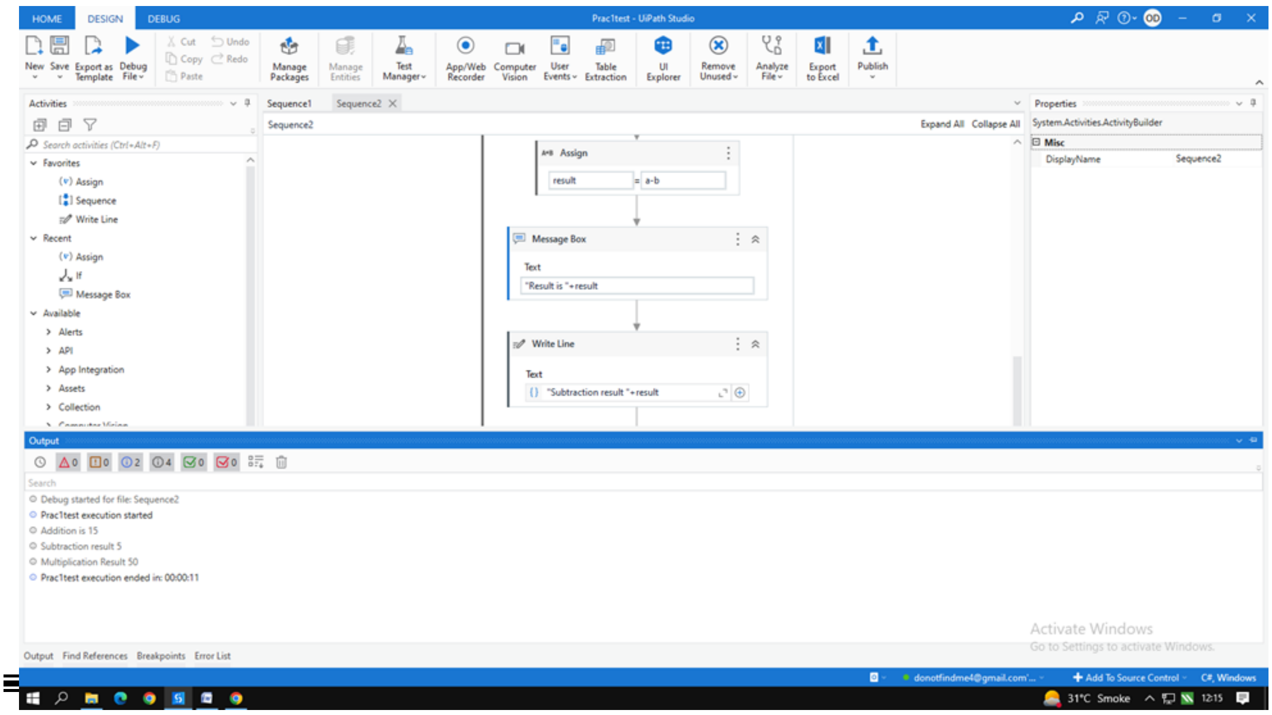
Message Box
Text
"Result is "+result

Assign
result = a-b

Message Box
Text
"Result is "+result

Write Line
Text
"Subtraction result "+result

Assign
result = a*b



Assign
result = a-b

Message Box
Text
"Result is "+result

Write Line
Text
"Subtraction result "+result

```
Debug started for file: Sequence2
Prac1test execution started
Addition is 15
Subtraction result 5
Multiplication Result 50
Prac1test execution ended in: 00:00:11
```

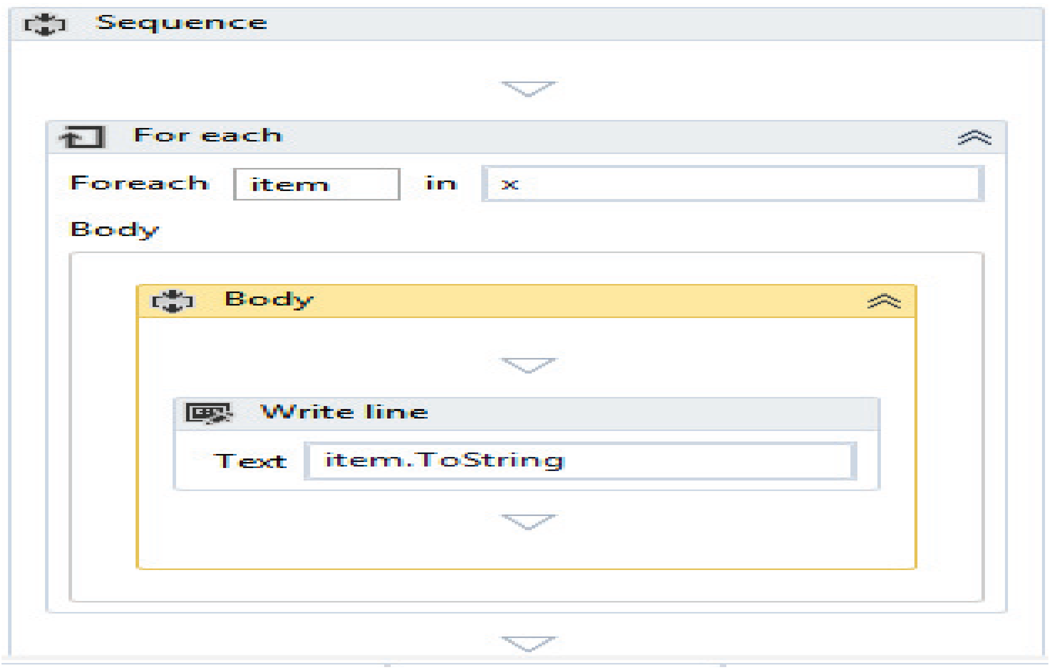
Practical No. 2

Aim : Create an automation UiPath project to calculate sum of an array.

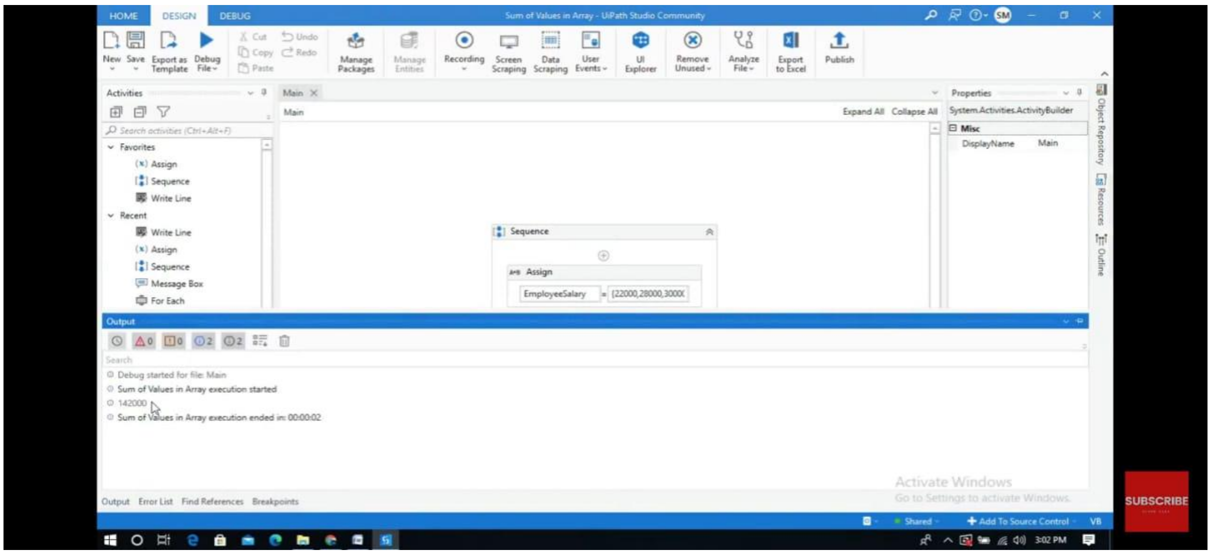
Steps:

1. Start with a Blank project in UiPath.
2. Add a Sequence activity to the Designer panel.
3. Next, add a For each activity within the Sequence and create an integer type array variable, x.
4. In the default value of the variable, put in {2,4,6,8,10,12,14,16,18,20}.
5. Create a variable sum and write it's default as 0.
6. In the body drag and drop assign activity and assign sum = sum + item in it
7. Outside the foreach loop drag and drop writeline activity and write sum in it.

Output:



In the body we also have to add an assign activity and in it **assign sum = sum + item** before write line


Sum of Values in Array - UiPath Studio Community
HOME DESIGN DEBUG
Sequence
Assign
EmployeeSalary = {22000,28000,3000(
Output

```
Debug started for file: Main
Sum of Values in Array execution started
142000
Sum of Values in Array execution ended in: 00:00:02
```

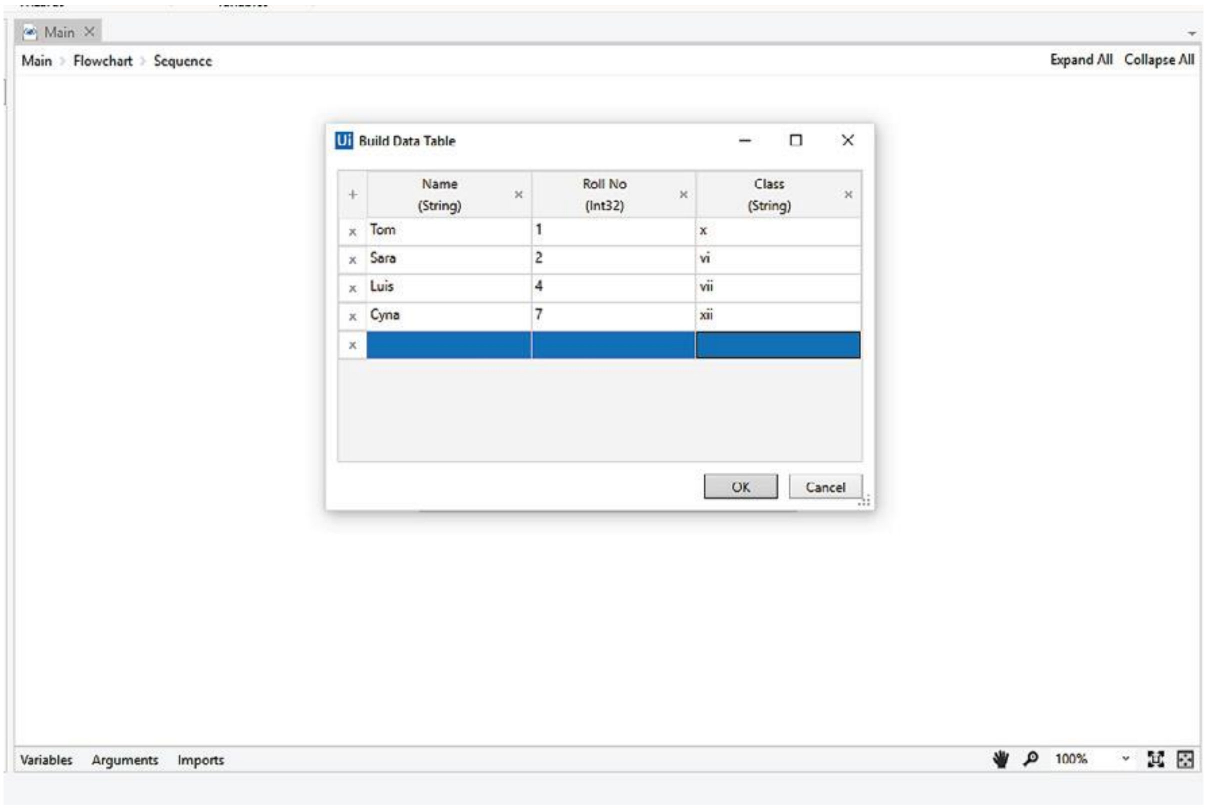
Practical No. 3

Aim : Create an automation UiPath project to create and display a data table.

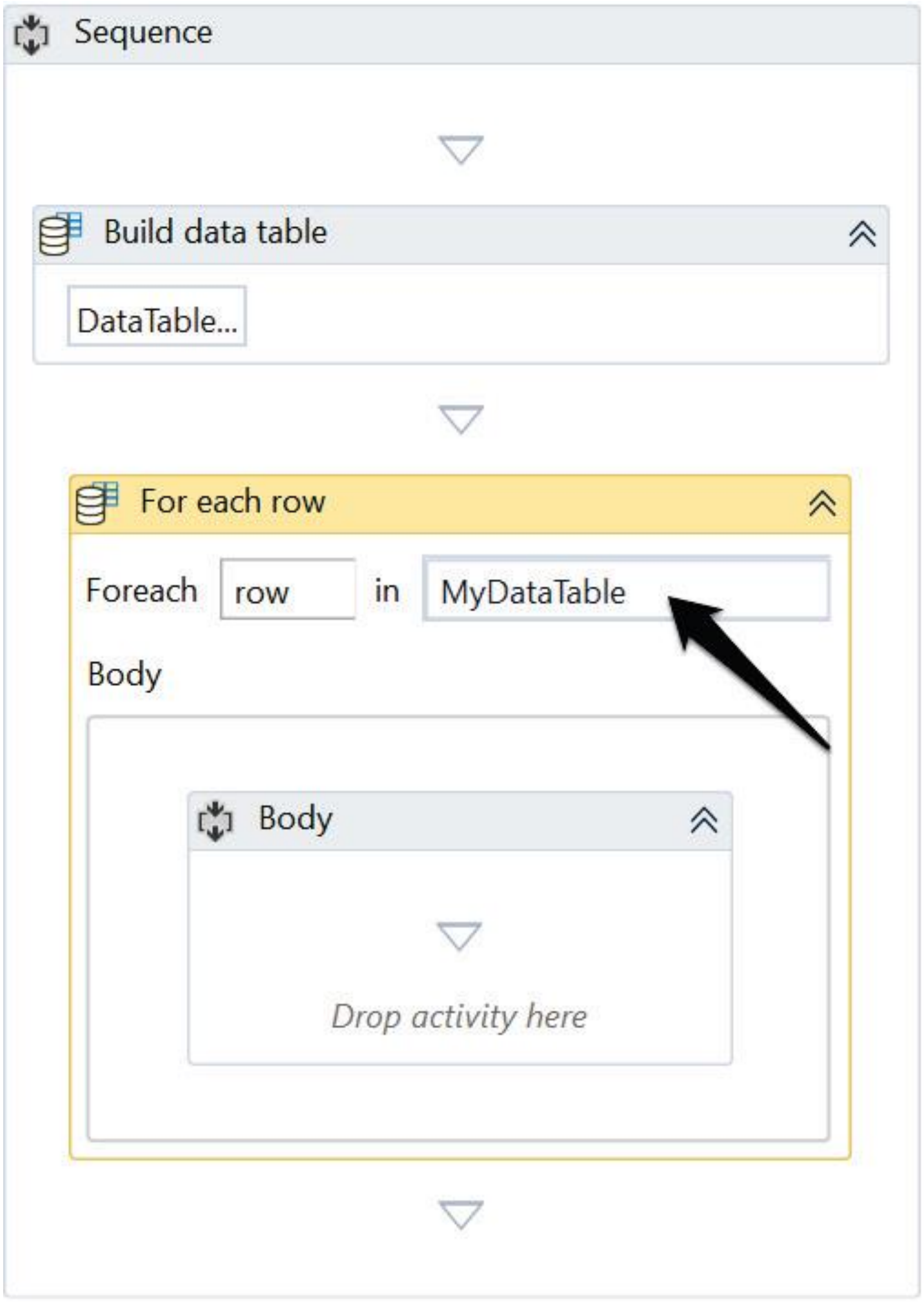
Steps:

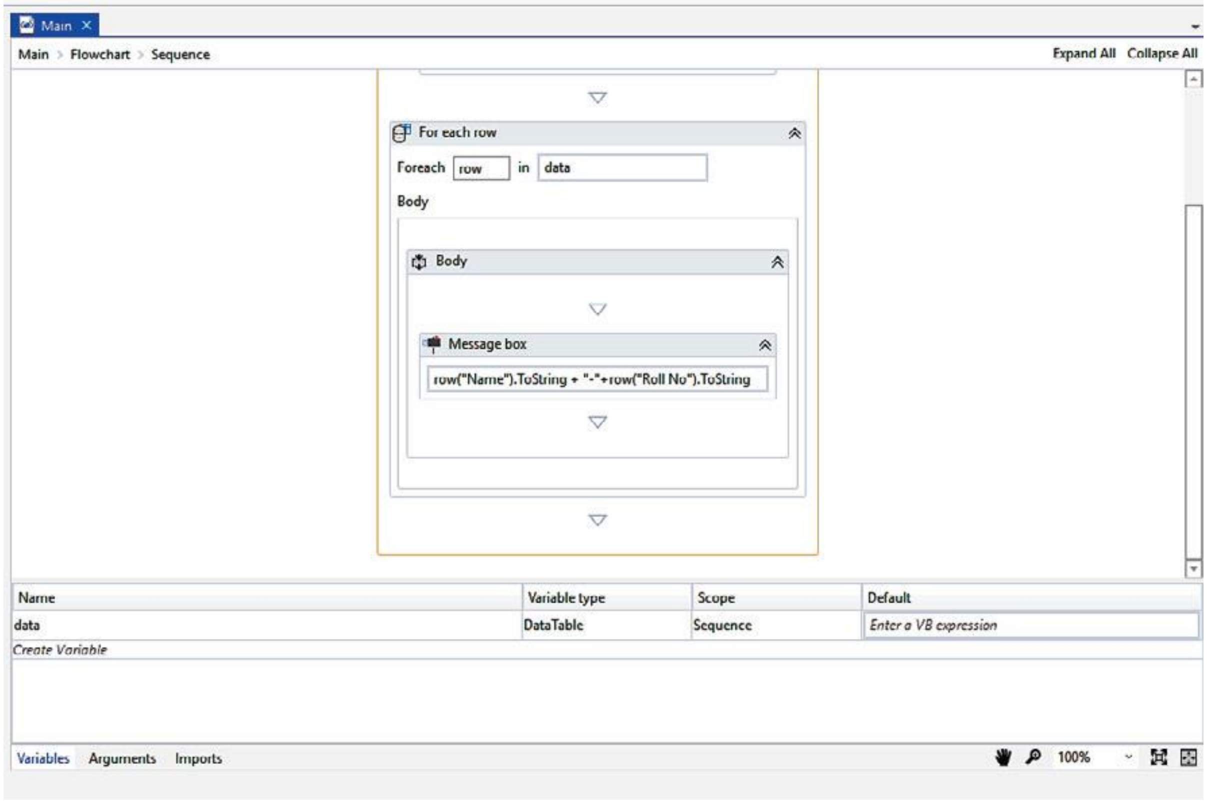
1. Drag and drop a Flowchart activity on the Designer panel. Also, drag and drop a Sequence activity and set it as the Start node.
2. Double click on the Sequence and drag and drop the Build Data Table activity inside the Sequence activity.
3. Click on the **Data Table** button. A pop-up window will appear on the screen. Remove both the columns (auto generated by the **Build Data Table** activity) by clicking on the Remove Column icon.
4. Now, we will add three columns by simply clicking on the + symbol. Specify the column names and select the appropriate data types from the drop-down list. Click on the OK button. We will add column Name of String Data Type,Roll_no of Int32 type and finally Class of string type. Now enter some random values just to insert the data into the rows. Click on the OK button and our data table is ready. We have to iterate over the data table's rows to make sure everything works correctly.
5. In order to store the Data Table created by Build Data Table activity, we have to create a data table variable mydatatable of DataTable type and in order to store the result of the data table that we have dynamically built. Also, specify assign the Output property of the Build Data Table activity with this variable. Specify the data table variable's name there.
6. After our data table is ready, we will iterate the data table's rows to make sure everything works correctly. Drag and drop the For each row activity from the Activities panel inside the Sequence activity. Specify the data table variable'sname in the expression text box of the For each row activity
7. Drag and drop a Message box activity inside the For each row activity. In the Message box activity, Inside the message box we have to write following string: row ("Name").ToString +"-"+ row("Roll_No").ToString + "- "+row("Class").ToString. row the variable which holding data for the data row in each iteration

Output:



Name (String)	Roll No (Int32)	Class (String)
Tom	1	x
Sara	2	vi
Luis	4	vii
Cyna	7	xii


Sequence
Build data table
DataTable...
For each row
Foreach
row
in
MyDataTable
Body
Body
Drop activity here



Foreach row in data

```
row("Name").ToString + "-"+row("Roll No").ToString
```

Name	Variable type	Scope	Default
data	DataTable	Sequence	*Enter a VB expression*

Create Variable

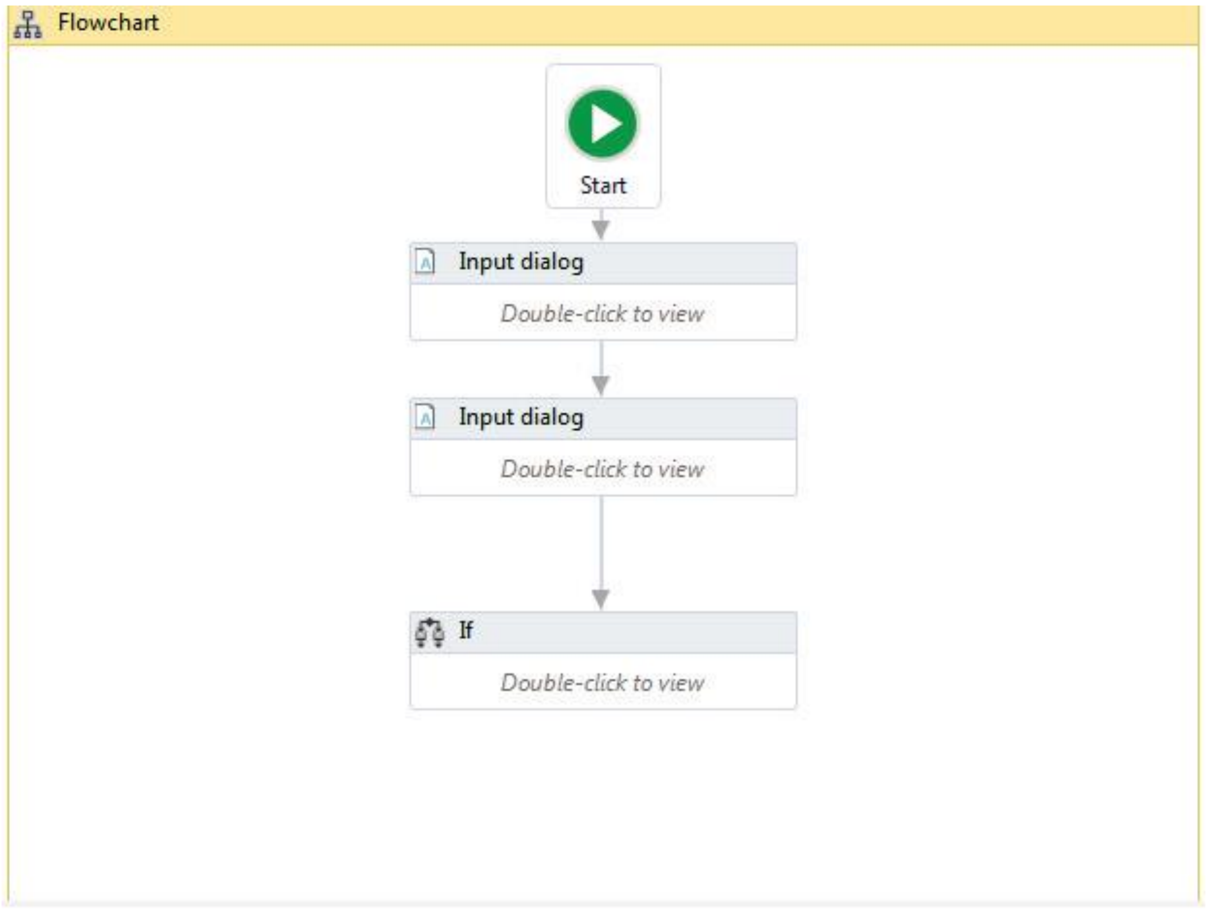
Practical No. 4

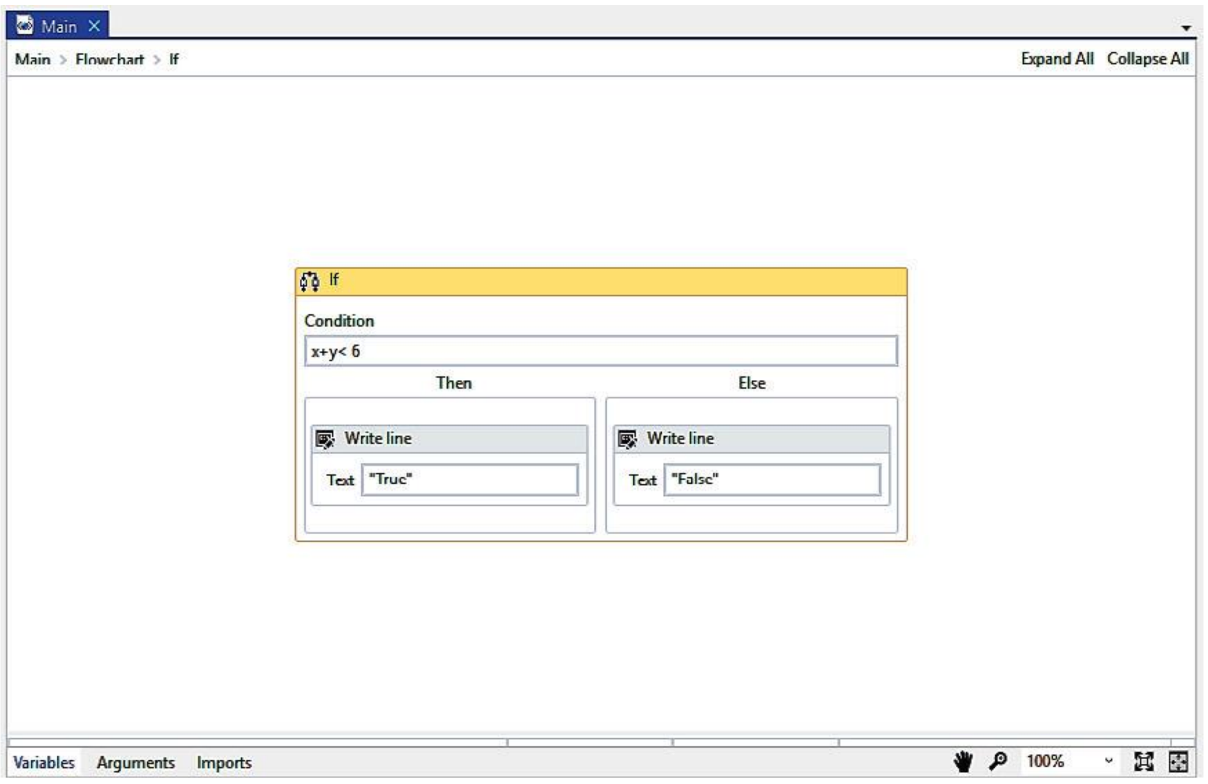
Aim : Create an automation UiPath Project using if-else statements.

Steps:

1. Add a Flowchart from the Activities panel.
2. Add two Input dialog activities. Create two integer variables, x and y.
3. In the Properties panel, change the label name and title name of both the Input dialog activities.
4. Now, specify these name of these two variables in the Result property of both the Input dialog activities.
5. Now add the If activity to the Designer panel
6. In the condition part, x+y<6, check whether it is true or false. Add two Write line activities and type "True" in one and "False" in the other
7. Click the Run button to check the output. If the condition holds true then it will show the true value; otherwise, it will show the false value, as shown in the second screenshot (in our case, we put in the values of x and y as 9 and 4, respectively, thus getting a sum of 13, which is not less than 6; hence, the output shows it as false value)

Output:





Condition
x+y< 6

Then

Write line
Text "Truc"

Else

Write line
Text "Falsc"

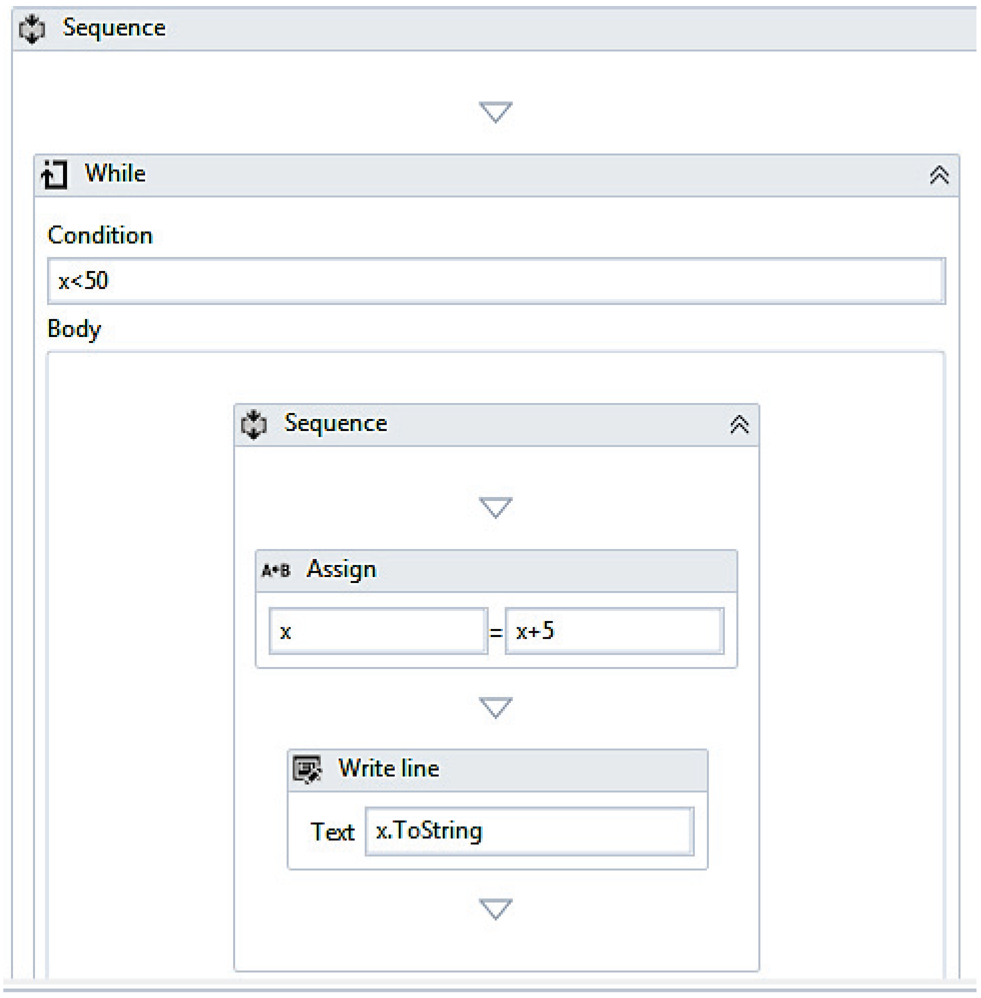
Practical No. 5

Aim : Create an automation UiPath Project using while statements.

Steps:

1. On a Blank project, add a Sequence activity.
2. Now, create an integer type variable x. Set its default value to 5.
3. Next, add a While activity to the Sequence.
4. In the condition field, set x<50.
5. Add an Assign activity to the body section of the While loop.
6. Now, go to the Properties panel of the Assign activity and type in the text field integer variable for value field integer x+5.
7. Drag and drop a Write line activity and specify the variable name x and apply ToString method on this variable

Output:



```
While
Condition: x<50
Body:
  Sequence
    Assign: x = x+5
    Write line: Text x.ToString
```

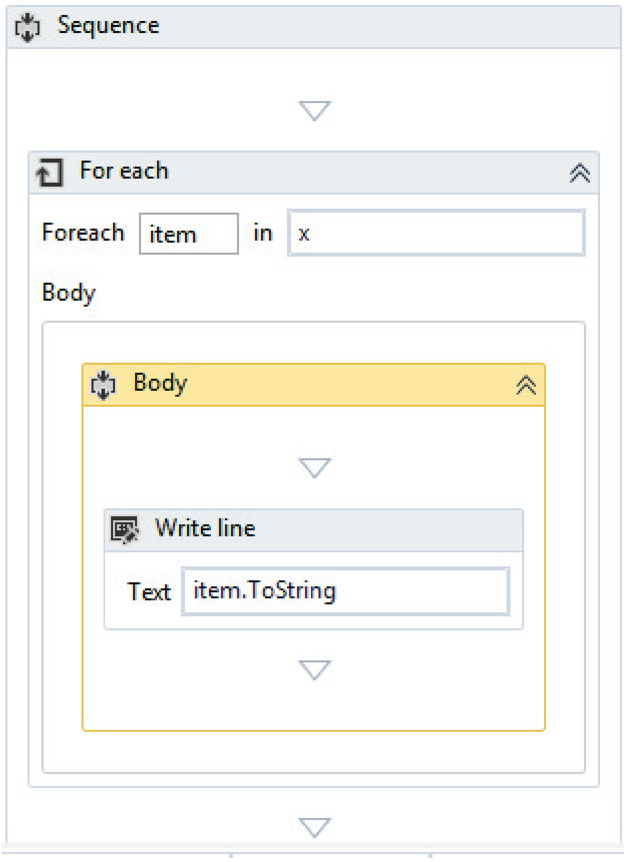
Practical No. 6

Aim : Create an automation UiPath Project using foreach statements.

Steps:

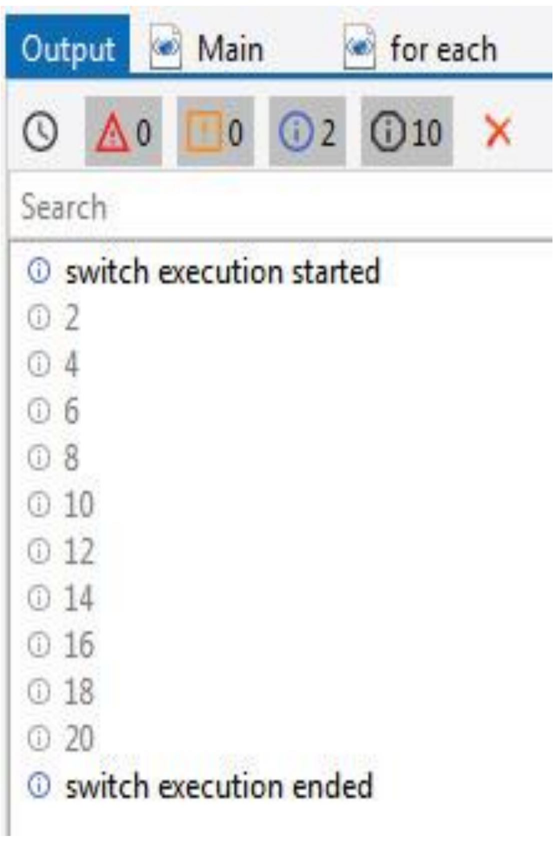
1. Start with a Blank project in UiPath.
2. Add a Sequence activity to the Designer panel.
3. Next, add a For each activity within the Sequence and create an integer type array variable, x.
4. In the default value of the variable, put in {2,4,6,8,10,12,14,16,18,20}.
5. Add a Write line activity to the Designer Panel (this activity is used to display the results).
6. In the Text field of the Write line activity, type item.ToString to display the output

Output:



Foreach item in x

Text item.ToString



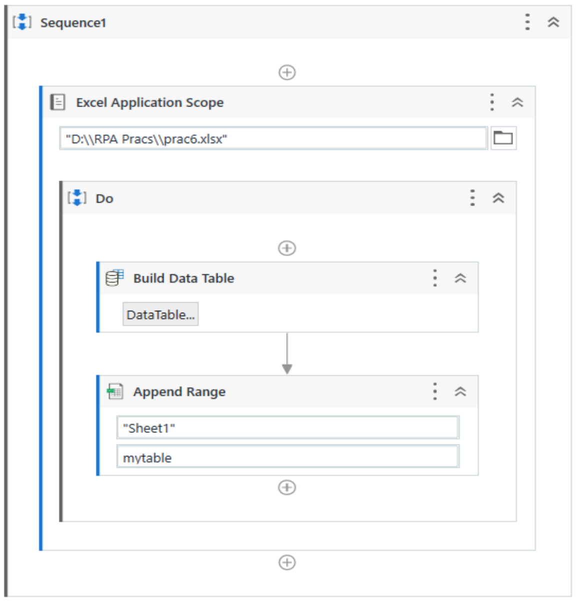
```
switch execution started
2
4
6
8
10
12
14
16
18
20
switch execution ended
```

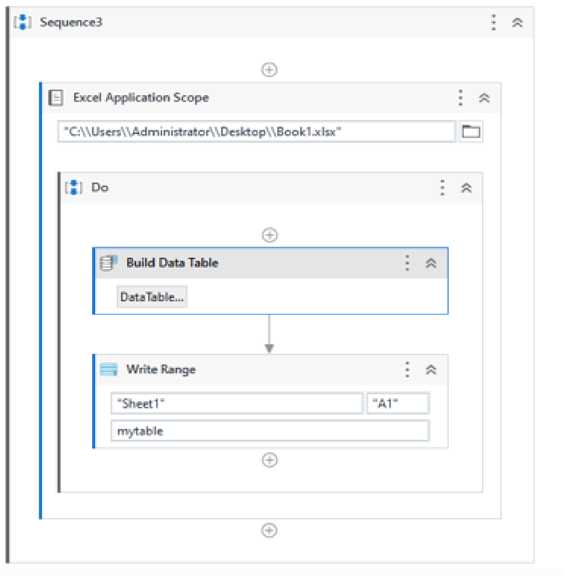
Practical No. 7

Aim : Automate the process to extract data from an excel file into a data table and vice versa

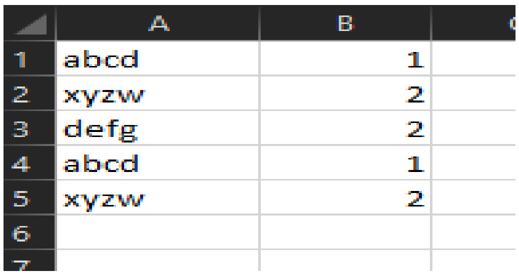
Steps:

1. Create a new sequence.
2. Create a excel file and keep the file empty
3. Drag and drop Excel application scope from the activities panel
4. In the application scope, browse the excel file in the system and add the path of the file.
5. Drag and drop Build data table activity and add a datatable and click on create option
6. On the right side of the page, right click on the output option and create a variable "mytable"
7. In the DO panel of the excel application scope, Drag and drop append range / write range activity.
8. It will automatically take the sheet as "Sheet1" and cell as "A1" as the initial point if you use write range.
9. It will also give a text box to enter the data you wish to write in the excel file. Add the variable name as mytable.
10. Run the automation and check the output.





Output: This automation will write the data of the data table in the excel file and append the data the number of times you run the automation.



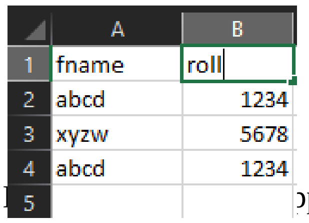
	A	B
1	abcd	1
2	xyzw	2
3	defg	2
4	abcd	1
5	xyzw	2
6		

Practical No. 8

Aim : Create an application automating the read, write and append operation on excel file.

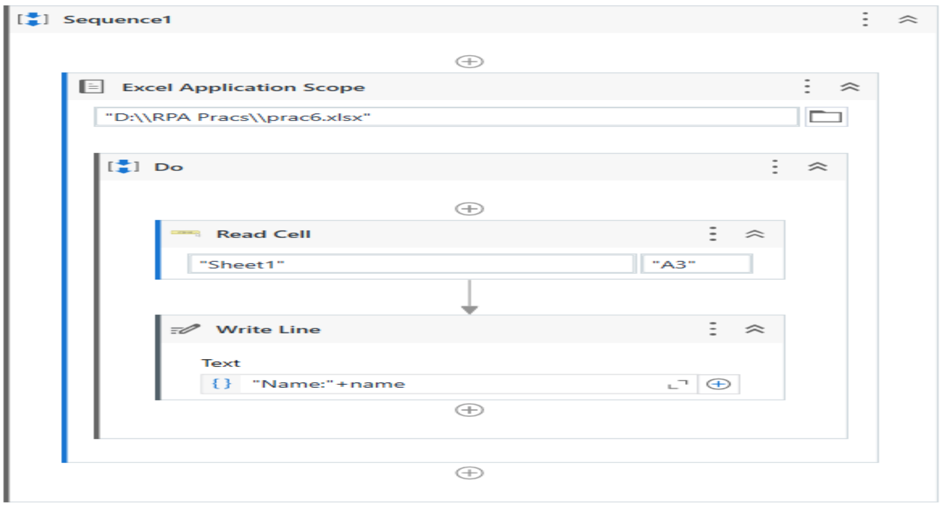
Steps:

1. Create a new sequence.
2. Create a excel file and add 2 columns of data



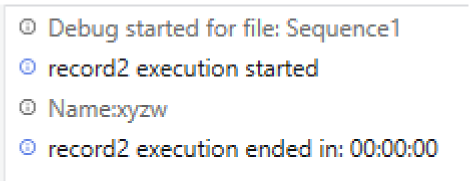
	A	B
1	fname	roll
2	abcd	1234
3	xyzw	5678
4	abcd	1234
5		

3. I pplication scope from the activities panel
4. In the application scope, browse the excel file in the system and add the path of the file.
5. In the DO panel of the excel application scope, Drag and drop read cell activity.
6. It will automatically take the sheet as “Sheet1” and cell as “A1” as the initial point .
7. Right click on the “Sheet1” and select create variable and add a variable as name and add any cell number you want to print.
8. Drag and drop a Writeline activity and add : “Name:”+name
9. Run the automation and check the output.



Sequence1
Excel Application Scope
"D:\\RPA Pracs\\prac6.xlsx"
Do
Read Cell
"Sheet1" "A3"
Write Line
Text
"Name:"+name

Output: The system will print the data available in cell A3



```
Debug started for file: Sequence1
record2 execution started
Name:xyzw
record2 execution ended in: 00:00:00
```

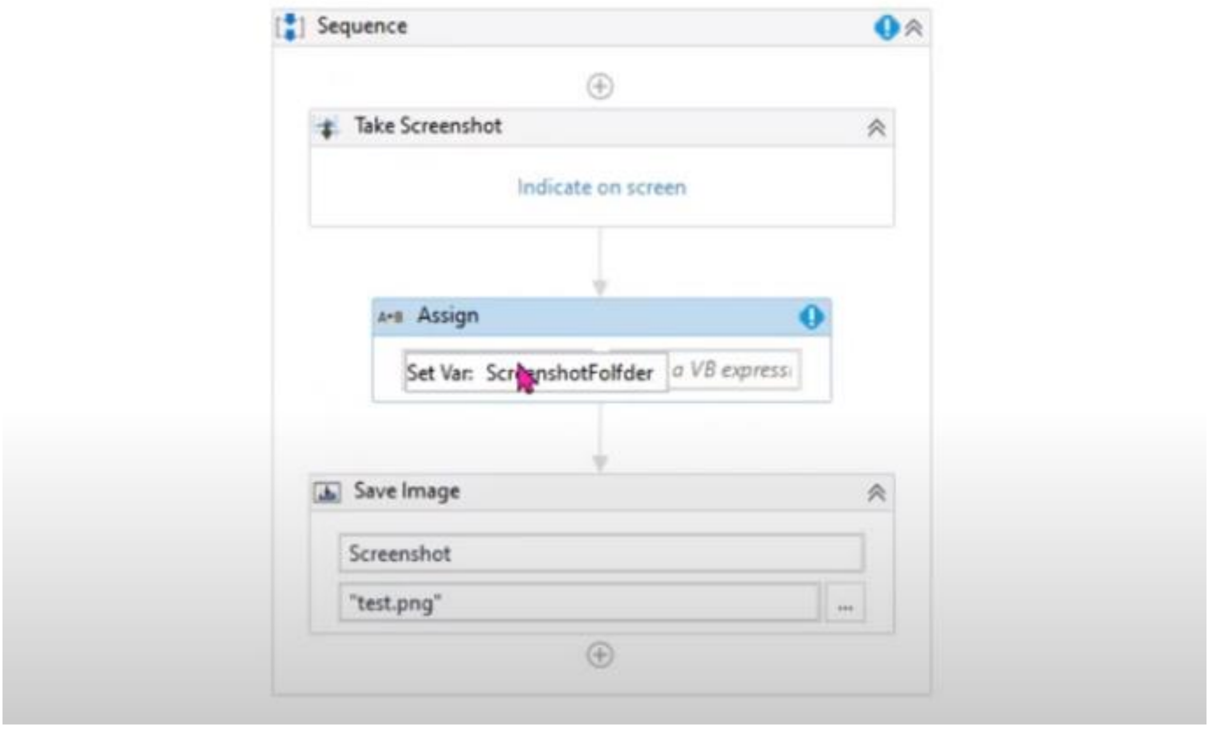
Practical No. 9

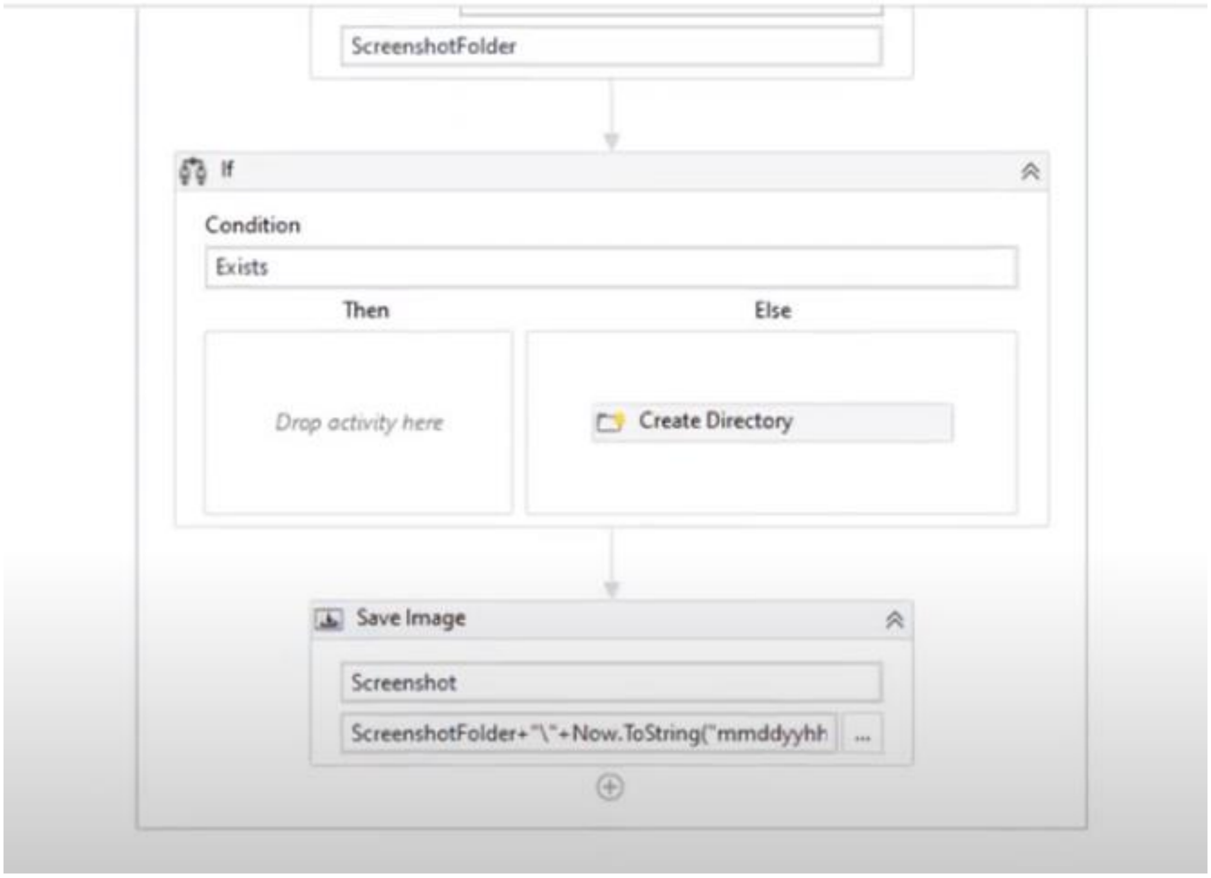
Aim : Automate the process of taking screenshots in UiPath.

Steps:

1. Create a new sequence.
2. Determine the specific point or event in your workflow where you want to capture a screenshot. This could be after a particular action, when an error occurs, or any other relevant moment.
3. Add the "Take Screenshot" Activity
4. Set the Selector Property: Define the screen region or UI element you want to capture. You can interactively select it using the "Indicate on Screen" feature or manually enter the selector using the Selector Editor
5. Set the OutputImage Property: Create a variable of type System.Drawing.Bitmap to store the captured screenshot. This variable will hold the screenshot image.
6. If you want to save the screenshot as an image file, use the "Save Image" activity. Configure it to save the screenshot to a specific file path.
7. Save and Run the Project

Output:




ScreenshotFolder

Exists

Screenshot
ScreenshotFolder+"\"+Now.ToString("mmddyyhh

Practical No. 10

Aim : Demonstrate the following events in UiPath:

i. Element triggering event

ii. Image triggering event

1. Open Studio and create a new **Process** named by default **Main**.
2. Drag a **Sequence** container in the **Workflow Designer**.
3. Create the following variable:

Variable Name	**Variable Type**
ContinueMonitor	**Boolean**

4. Drag a **Log Message** activity inside the **Sequence** container.
   - o In the **Properties** panel, select the **Level** option from the **Message** drop-down list.
   - o Add the expression "Start monitoring..." in the **Message** field.
5. Add an **Assign** activity under the **Log Message** activity.
   - o In the **Properties** panel, add the variable ContinueMonitor in the **To** field.
   - o Add the condition True in the **Value** field.
6. Place a **Monitor Events** activity below the **Assign** activity.
   - o In the **Properties** panel, add the value ContinueMonitor in the **RepeatForever** field.
7. Drag a **Hotkey Trigger** activity inside the **Monitor Events** activity. This activity opens the **Calculator** app from **Windows**.
   - o Select the checkboxes for the Alt and Shift options.
   - o In the Key field, type the letter c.
   - o In the **Properties** panel, select the option **EVENT_BLOCK** from the **EventMode** drop-down list.
8. Drag another **Hotkey Trigger** activity and place it next to the previous **Hotkey Trigger** activity. This activity opens a new browser tab and searches on Google the text previously selected by the user.
   - o Select the checkboxes for the **Alt** and **Shift** options.
   - o In the **Key** field, type the letter g.
   - o In the **Properties** panel, select the option **EVENT_BLOCK** from the **EventMode** drop-down list.
9. Drag another **Hotkey Trigger** activity and place it next to the previous **Hotkey Trigger** activity. This activity stops monitoring the events.
   - o Select the check boxes for the **Alt** and **Shift** options.
   - o In the **Key** field, type the letter s.
   - o In the **Properties** panel, select the option **EVENT_BLOCK** from the **EventMode** drop-down list.
10. Add a new **Sequence** container and place it below the **Hotkey Trigger** activity.

- In the **Properties** panel, add the name Event Handler in the **DisplayName** field.
- Create the following variable:

Variable Name	Variable Type
TriggerHotkey	**UiPath.Core.EventInfo**
ContinueMonitor	**Boolean**

11. Drag a **Log Message** activity inside the **Event Handler**.

- In the **Properties** panel, select the **Info** option from the **Level** drop-down list.
- Add the expression "Event triggered" in the **Message** field.

12. Drag a **Get Event Info** activity below the **Log Message** activity.

- In the **Properties** panel, add the variable TriggerHotkey in the **Result** field.
- Select the **UiPath.Core.EventInfo** option from the **TypeArgument** drop-down list.

13. Place a **Switch** activity below the **Get Event Info** activity. All **Hotkey Triggers** are described inside this activity and treated as cases.

- In the **Properties** panel, add the value TriggerHotkey.KeyEventInfo.KeyName.ToLower in the **Expression** field.
- Select the **String** option from the **TypeArgument** drop-down list.

14. Click the **Add new case** button from the **Switch** activity.

- Add the value c in the **Case value** field.

15. Place an **Open Application** activity and place it inside the **Case c** container. This represents the first **Hotkey Trigger** case that opens the **Calculator** app.

- In the **Properties** panel, add the expression "calc.exe" in the **Arguments** field.
- Add the expression "<wnd app='applicationframehost.exe' title='Calculator' />" in the **Selector** field.

16. Click the **Add new case** button from the **Switch** activity.

- Add the value g in the **Case value** field.

17. Drag a **Sequence** container and place it inside the **Case g** container. This represents the second **Hotkey Trigger** case that initiates a Google search for the previously selected text.

- In the **Properties** panel, add the name Google selected text in the **DisplayName** field.
- Create the following variable:

Variable Name	Variable Type
TextToSearch	**GenericValue**

18. Drag a **Delay** activity and place it inside the **Google selected text** sequence.

- In the **Properties** panel, add the value 00:00:00.5000000 in the **Duration** field.

19. Add a **Copy Selected Text** activity below the **Delay** activity.

- In the **Properties** panel, add the value True in the **ContinueOnError** field.
- Add the variable TextToSearch in the **Result** field.
- Add the value 2000 in the **Timeout (milliseconds)** field.

20. Drag an **If** activity under the **Copy Selected Text** activity.

- In the **Properties** panel, add the expression TextToSearch IsNot Nothing in the **Condition** field.

21. Place an **Open Browser** activity inside the **Then** box.

- In the **Properties** panel, select the **IE** option from the **BrowserType** drop-down list.
- Add the expression "www.google.com" in the **Url** field.
- Select the checkbox for the **NewSession** option. This starts a new session in the selected browser.

22. Place a **Type Into** activity inside the **Do** sequence.

- In the **Properties** panel, select the **Target** option from the **Target** drop-down list.
- Add the expression "<webctrl tag='INPUT' aaname='Search' />" in the **Selector** field.
- Select the **INTERACTIVE** option from the **WaitForReady** drop-down list.
- Add the variable TextToSearch in the **Text** field.
- Select the checkbox for the **Activate** option. This option brings the UI element to the foreground and activates it before the text is written.
- Select the checkbox for the **SimulateType** option. This option simulates the type using the technology of the target application.

23. Drag a **Send Hotkey** application below the **Type Into** activity.

- In the **Properties** panel, add the expression "enter" in the **Key** field.
- Select the **Target** option from the **Target** drop-down list.
- Add the expression "<webctrl tag='INPUT' aaname='Search' />" in the **Selector** field.
- Select the **INTERACTIVE** option from the **WaitForReady** drop-down list.
- Select the checkbox for the **Activate** option. This option brings the UI element to the foreground and activates it before the text is written.
- Select the **None** option from the **KeyModifiers** drop-down list.
- Select the checkbox for the **SpecialKey** option. This option indicates if the use of a special key in the keyboard shorcut.

24. Drag a **Message Box** activity in the **Else** container.

- In the **Properties** panel, select the **Ok** button from the **Buttons** drop-down list.
- Add the expression "Text could not be copied. Please try again." in the **Text** field.
- Select the checkbox for the **TopMost** option. This option always brings the message box to the foreground.

25. Click the **Add new case** button from the **Switch** activity.

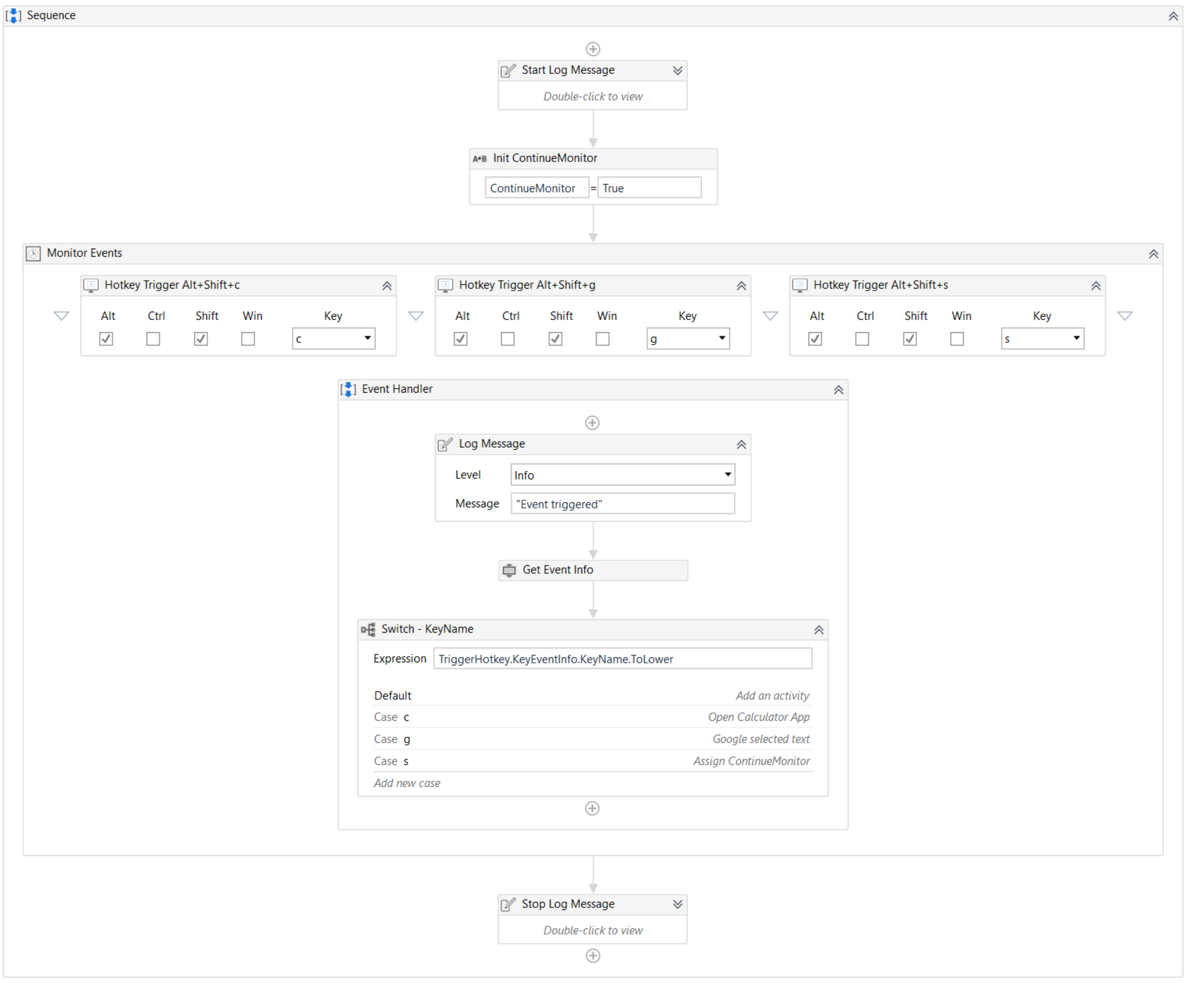
- Add the value s in the **Case value** field.

26. Drag an **Assign** activity inside the **Case s** container. This represents the third **Hotkey Trigger** case that stops monitoring the events.

- Add the variable ContinueMonitor in the **To** field.
- Add the condition False in the **Value** field.

27. Place a **Log Message** activity below the **Monitor Events** activity.

- In the **Properties** panel, select the **Info** option from the **Level** drop-down field.
- Add the expression "Stop monitoring.." in the **Message** field.



ContinueMonitor = True

Key: c / g / s

Level: Info
Message: "Event triggered"

Expression: TriggerHotkey.KeyEventInfo.KeyName.ToLower

Default	Add an activity
Case c	Open Calculator App
Case g	Google selected text
Case s	Assign ContinueMonitor
Add new case	

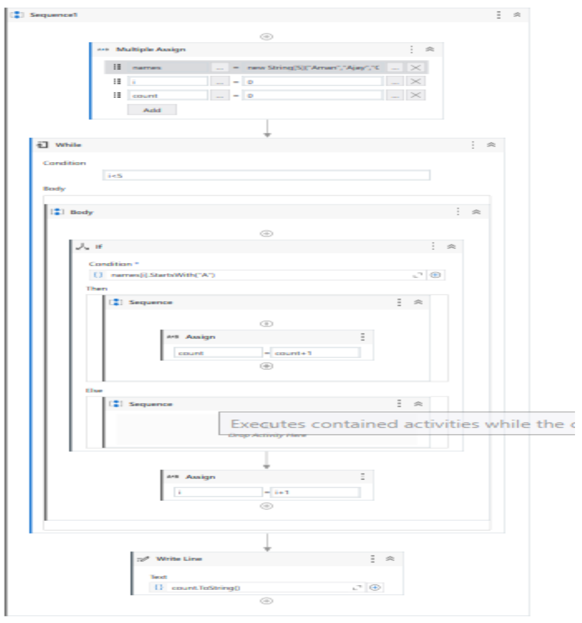
Practical No. 11

Aim : Consider an array of names. We have to find out how many of them start with the letter "a". Create an automation where the number of names starting with "a" is counted and the result is displayed.

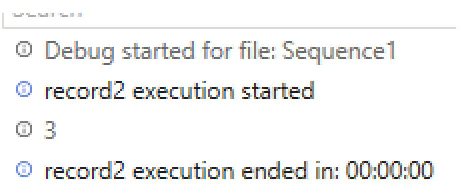
Steps:

1. Create a new sequence.
2. Drag and drop multiple assign activity
3. Create a string array and add the names you wish to add
   Names=New String[5]{“Aman”,”Ajay”,”Omkar”,”Ashish”,”Tejas”}
4. Once the array is created, go to the variable tab and change the data type of the array to Array of [T] and select the datatype as string.
5. Once array is created, create a new variable i=0 for iteration and count=0 and select the datatype as Int32 for both the variables.
6. Once done, Drag and drop a while activity from activities panel.
7. Add condition as i<5
8. In the body panel of the while activity, drag and drop an if activity
9. Add the following condition in the if activity:
   Names[i].StartsWith(“A”)
10. Drag and drop an assign activity in the “then” panel of the if activity and add Count=count+1(This will increase the counter by 1 if the condition is true)
11. After the else panel in the if activity, drag and drop an assign activity and add i=i+1
12. Outside the while loop in the main sequence, drag and drop writeline activity and add: Count.ToString().
13. Run the automation and check the output.

Output:



names = new String[5]{"Aman","Ajay","
i = 0
count = 0
i<5
names[i].StartsWith("A")
count = count+1
i = i+1
count.ToString()



```
Debug started for file: Sequence1
record2 execution started
3
record2 execution ended in: 00:00:00
```