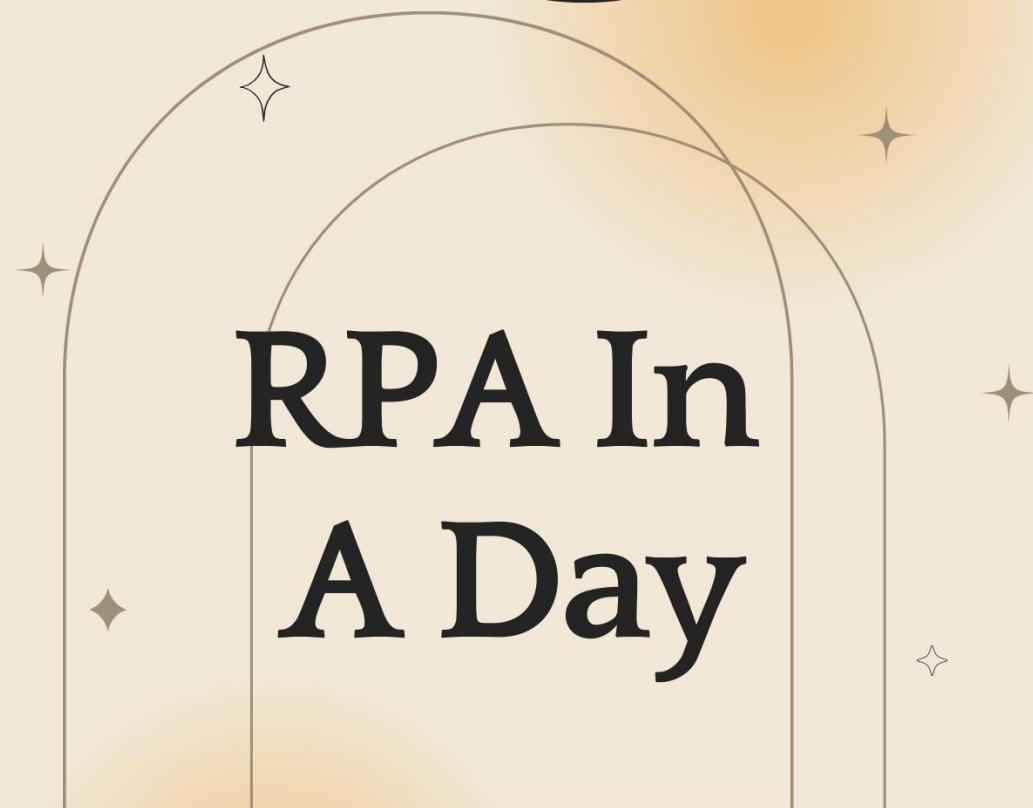




POWER
AUTOMATE
DESKTOP



RPA In A Day

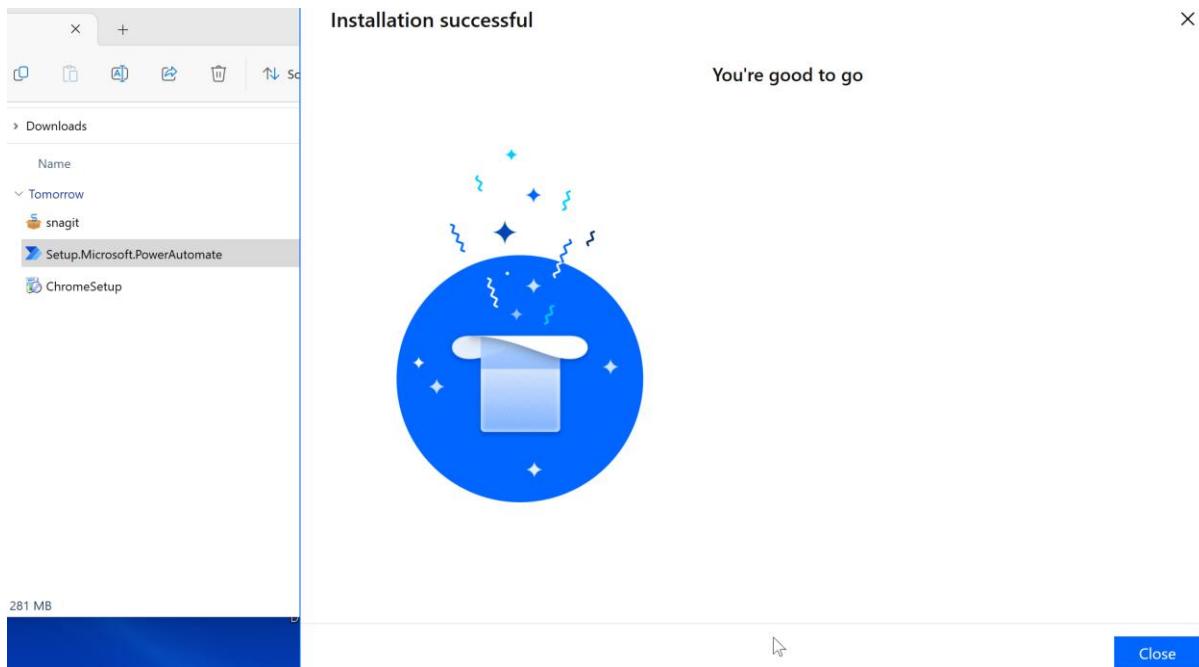
Written by Priyaranjan KS

Contents

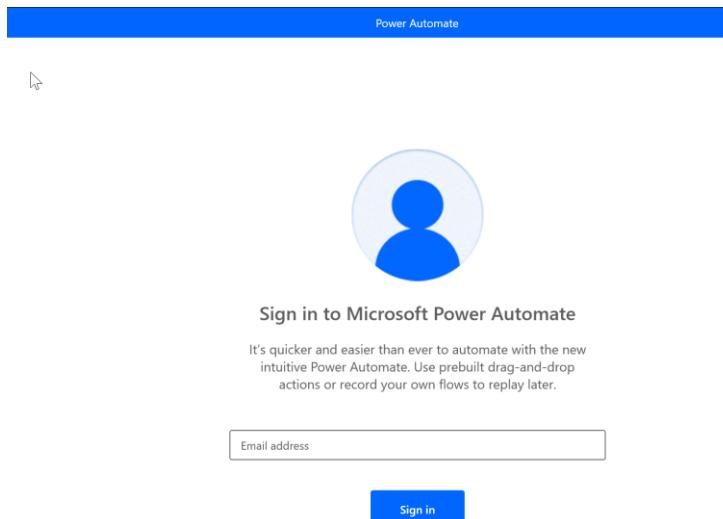
Power Automate Desktop Installation.....	3
PAD Exercise 1 - Variables.....	4
Add Text Variables	5
Dynamic Input Collection.....	10
Number variables & Convert Text to Number Action.....	16
Perform Variable Calculation	19
List Variable.....	20
Input/Output Variable	23
Mark Average Calculation.....	24
Run the Flow	24
PAD Exercise 2 – If , Else If and Loops.....	28
Loop Condition.....	30
If - Else If – Else Conditions	31
Boolean Variable.....	33
Run the flow.....	38

Power Automate Desktop Installation

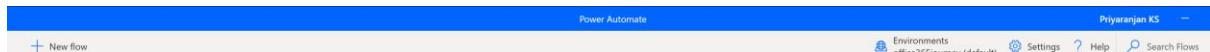
Download the MSI Installer file from [here](#). Once downloaded, run the installer. A successful installation will indicate the below message.



Search for the Power Automate App and open it . It will ask for the Work/School Email Address associated with the Office 365 for signing in . Provide the email address and sign in with the password.



It will open the App from which we can select the New Flow option to build a desktop flow.



Home My flows Shared with me Examples

Begin your automation journey

Get familiar with Power Automate and begin your automation journey.

 Get started tour
Get a tour on Power Automate for desktop and access all features available to elevate your experience.

[Start tour](#)

 Build desktop flows
Drag and drop actions to perform the steps of your automation and form a flow.

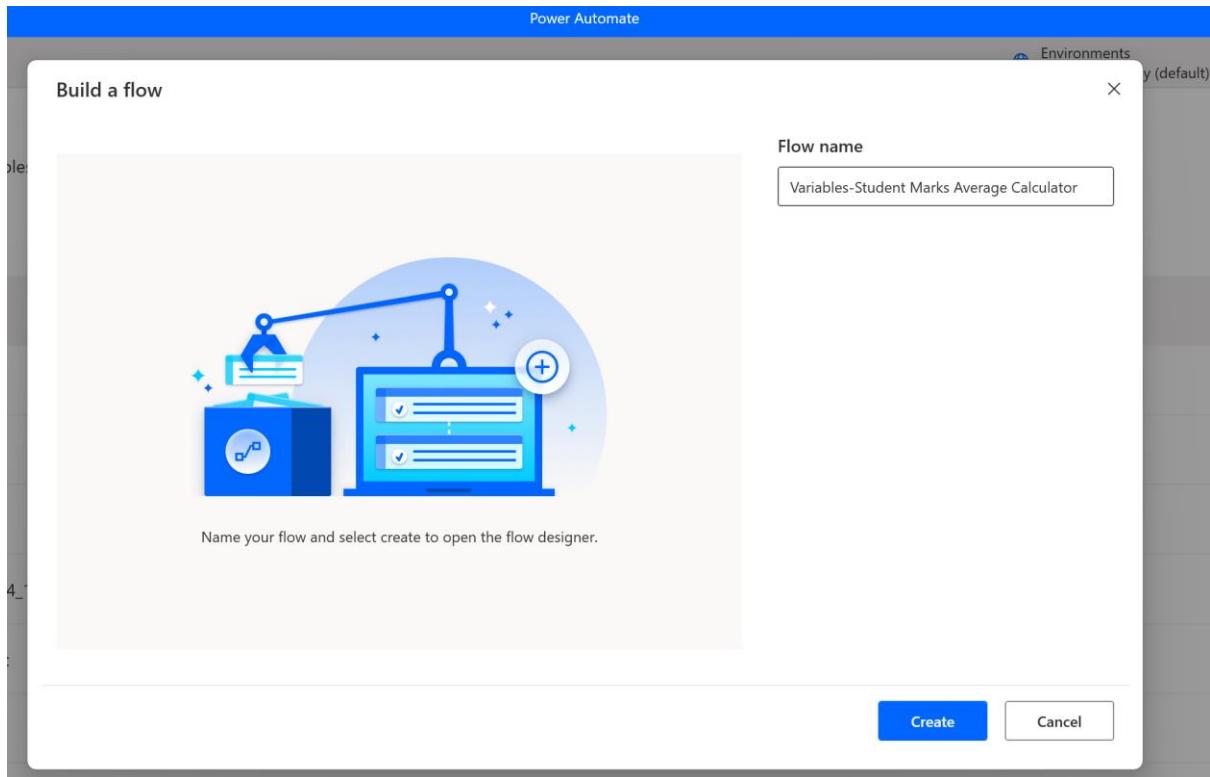
[+ New flow](#)

[Start with an example](#) [See all](#)

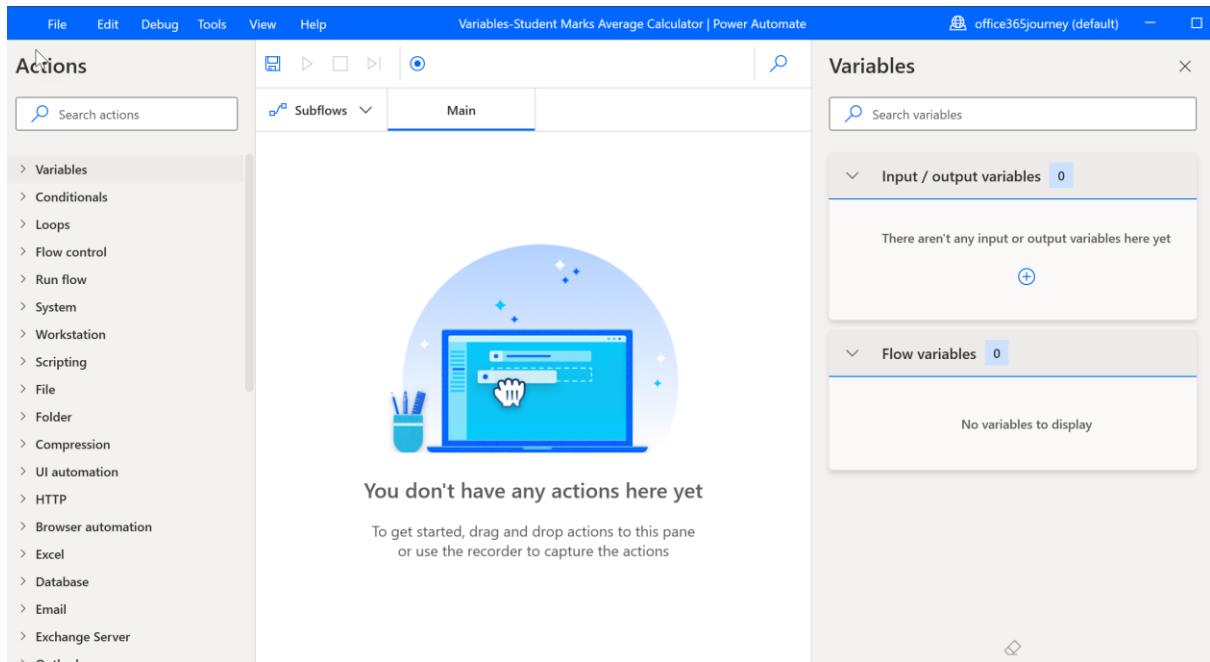
It will open up the Flow creation Studio where we can name the flow and click on create.

PAD Exercise 1 - Variables

Lets name it as *Variables-Student Marks Average Calculator*



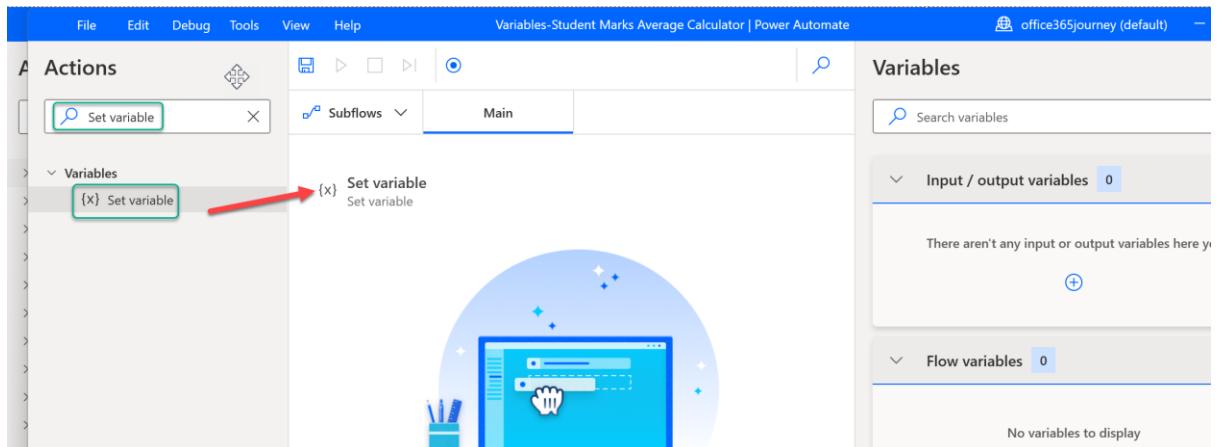
The blank studio for the new flow will look like below



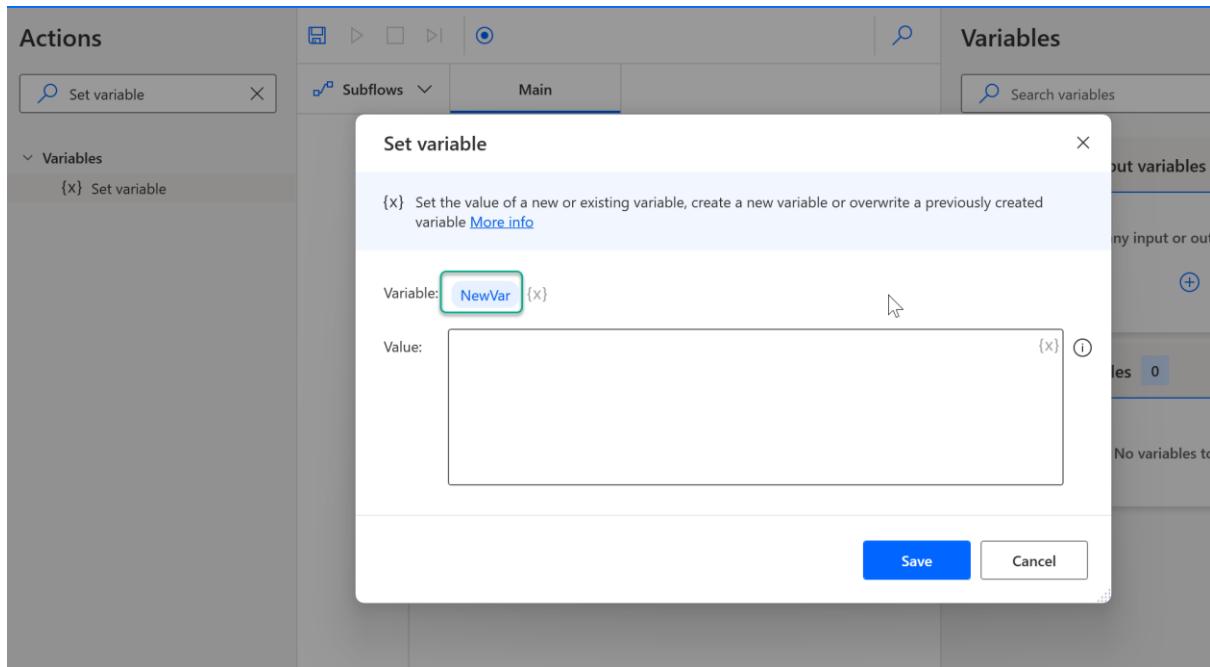
Add Text Variables

Lets add few variable steps to capture the data from user. We will initially add 3 Text variables to store the Subject names

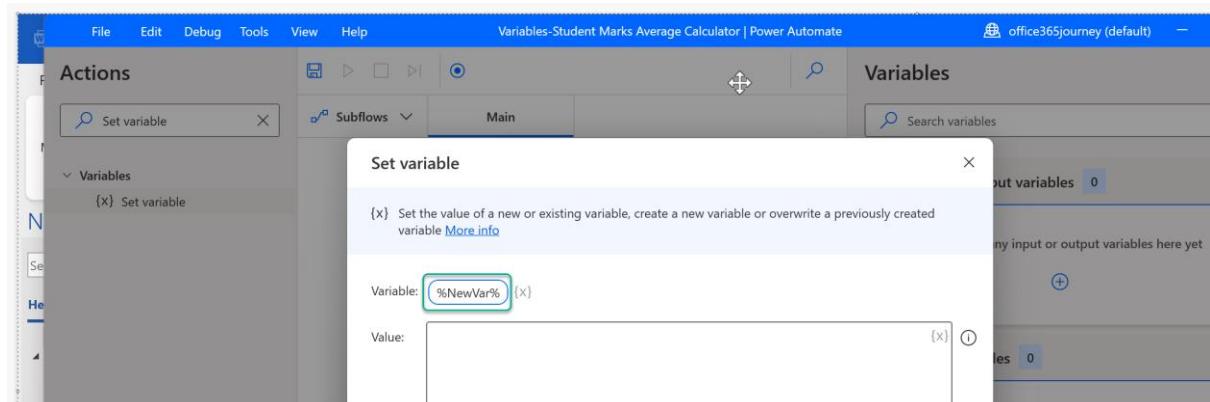
We will use the Set variable action to set the text variables. Search for the *Set Variable* action drag and drop it to the design studio



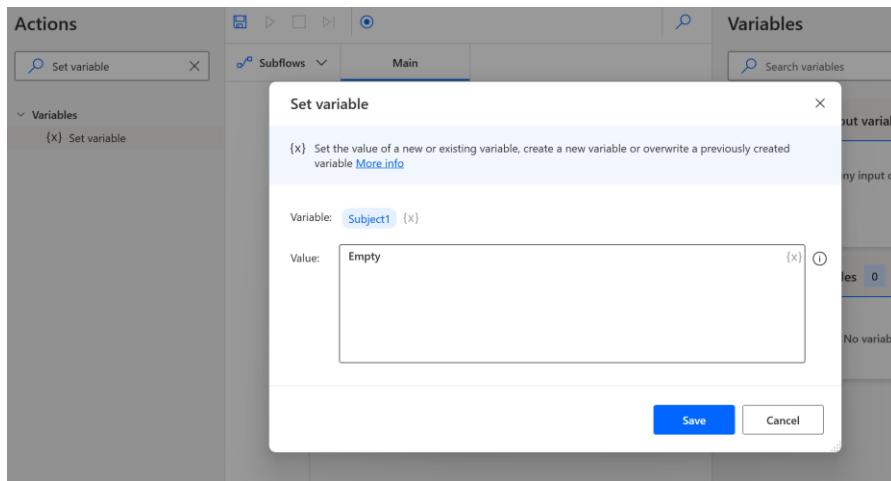
Click on the NewVar variable to rename the variable.



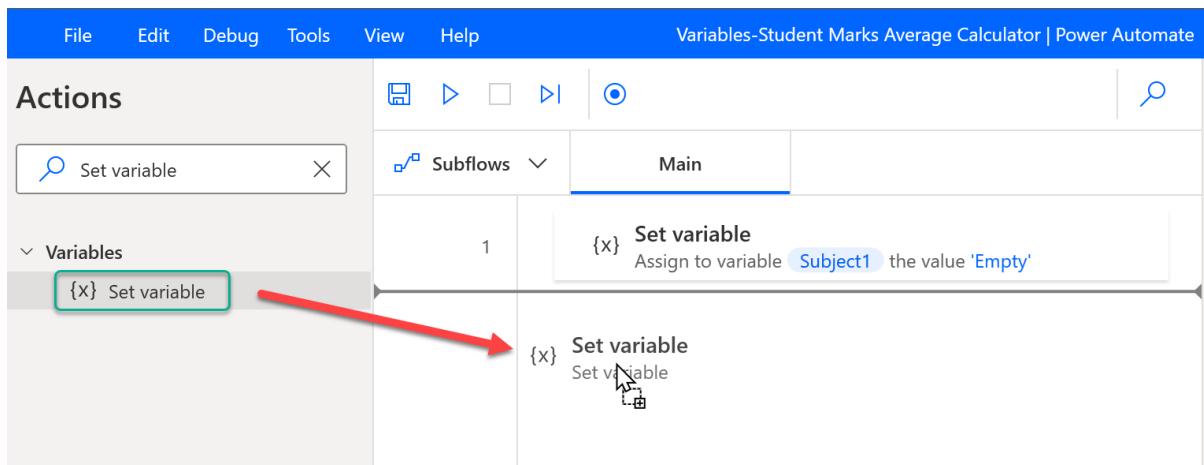
We can now see that it shows the variables name between 2 percentage symbols which is how variables are denoted in Power Automate Desktop. Even if we remove the complete name including percentage and type in a new name, the renamed variable will get saved with leading and trailing percentage symbols.



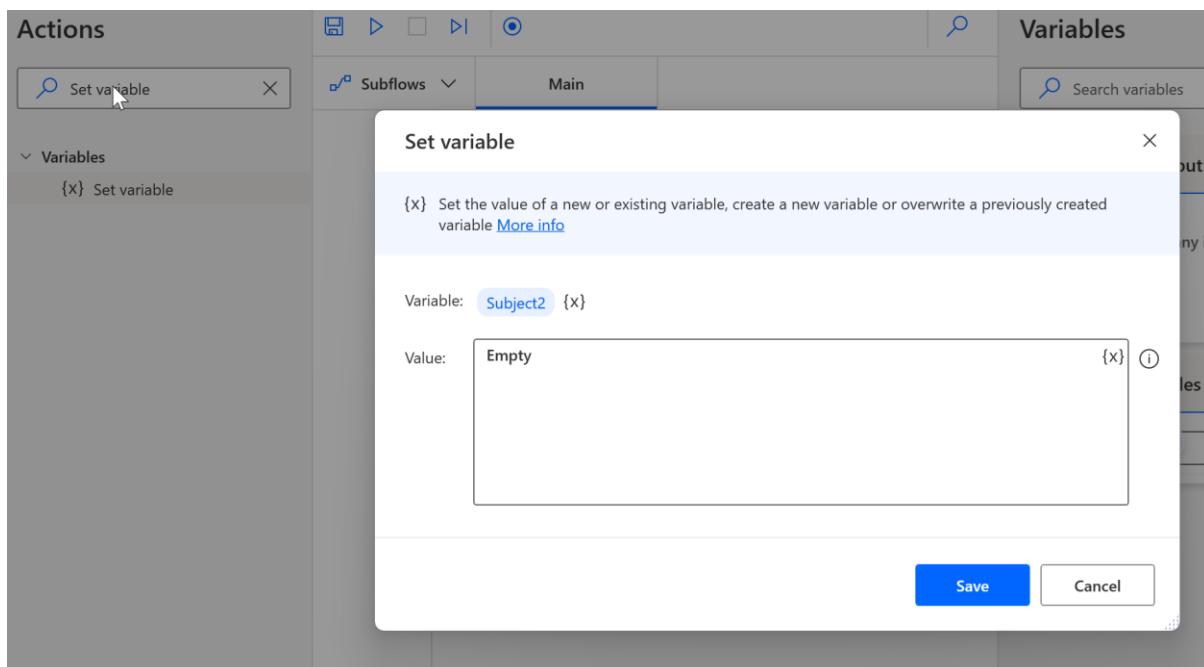
Lets rename it to Subject 1 and add a dummy value of *Empty* click on Save



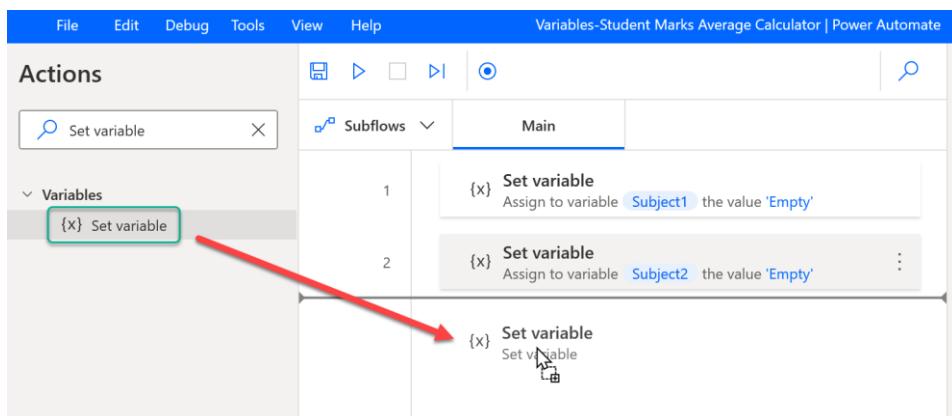
Lets drag and drop one more Set variable action to capture the second subject name



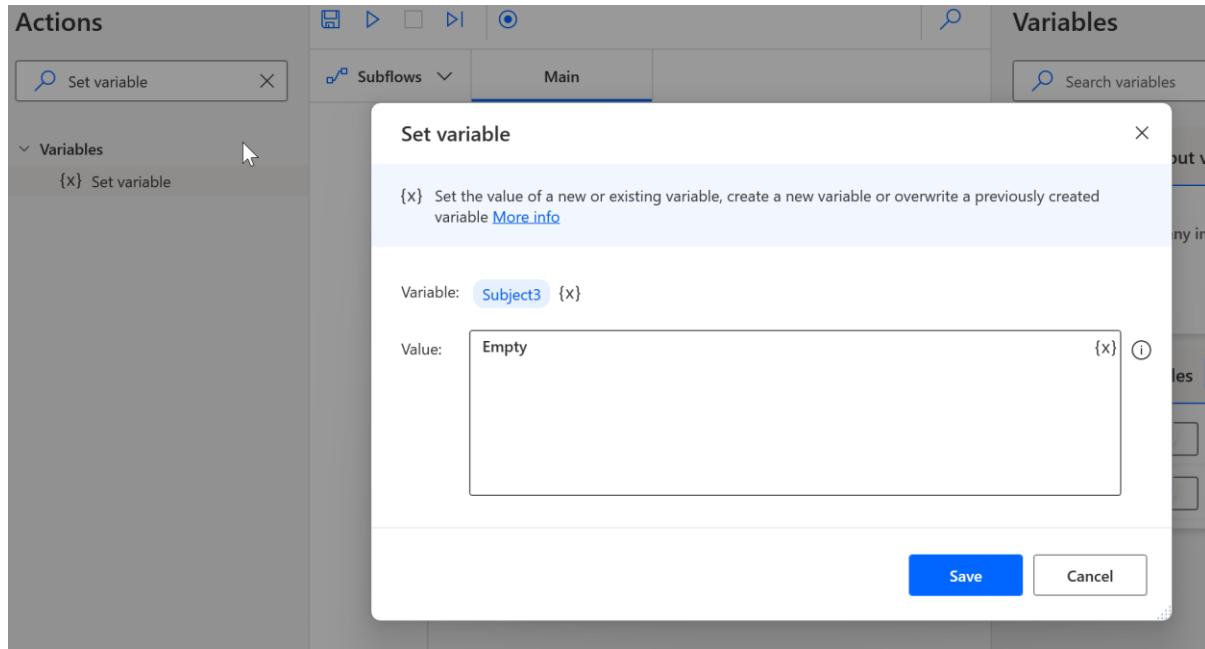
We will rename the variable to *Subject 2* and add the dummy value as Empty and click Save



Let's add one more Text variable to store the third subject name

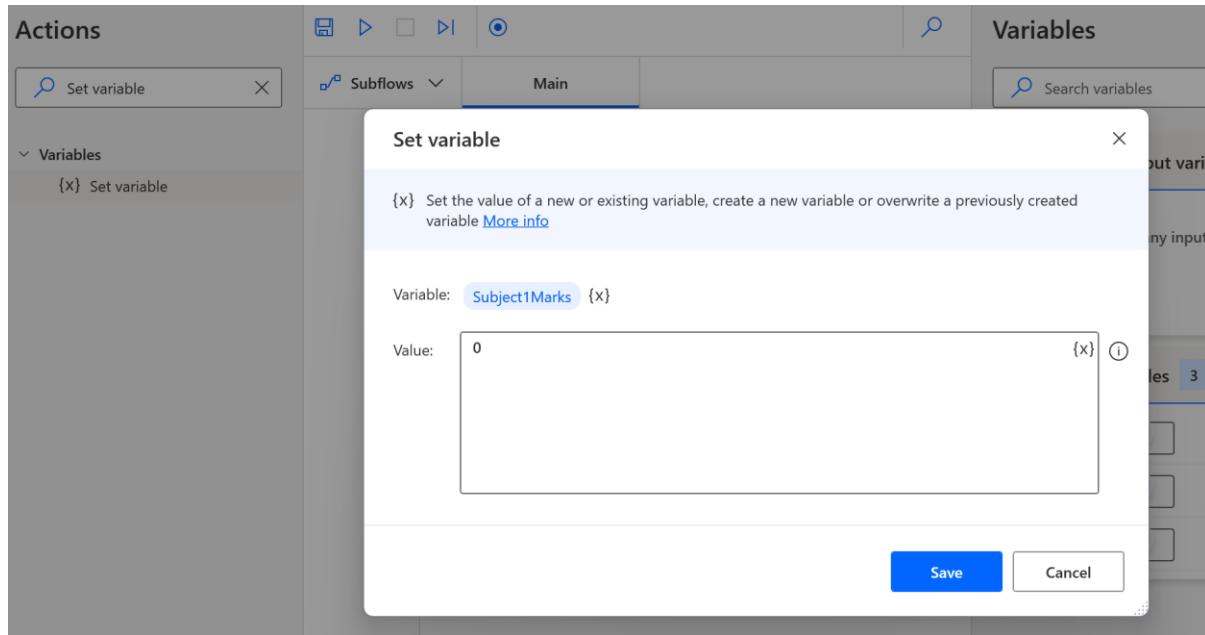


Rename the variable to Subject , add a dummy Empty value which we will replace with dynamic user input later. Click on Save

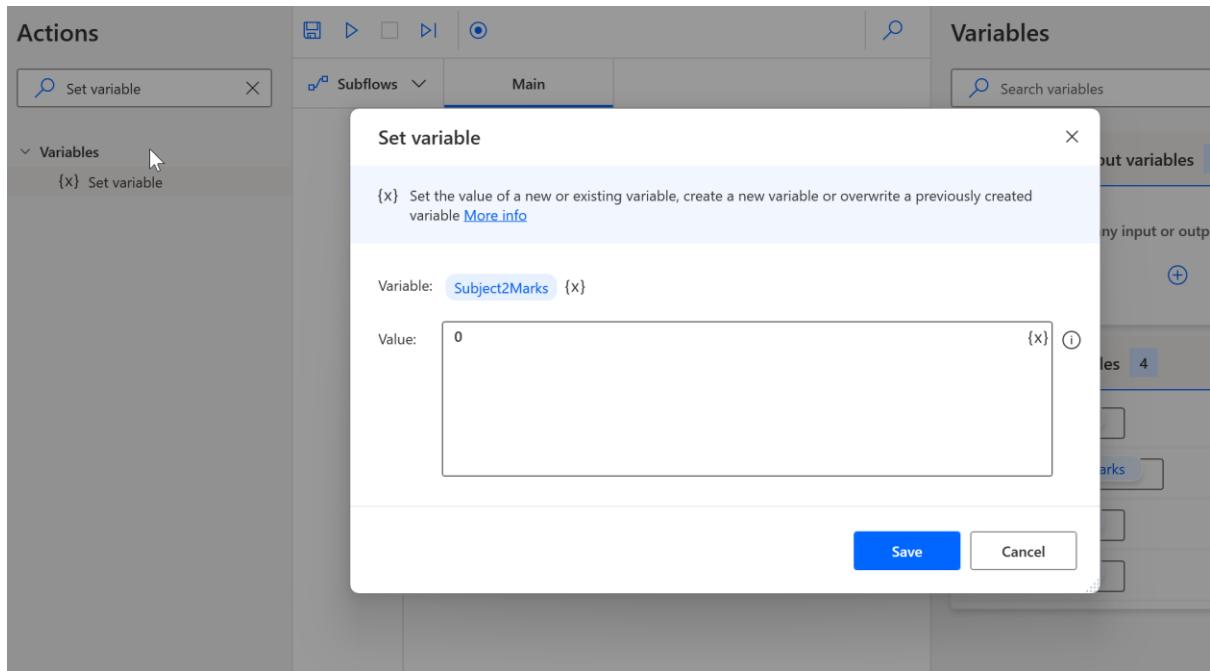


Now lets add variables to hold the Subject 1 Marks. This will be stored as a text value. We will see later how to convert this to number.

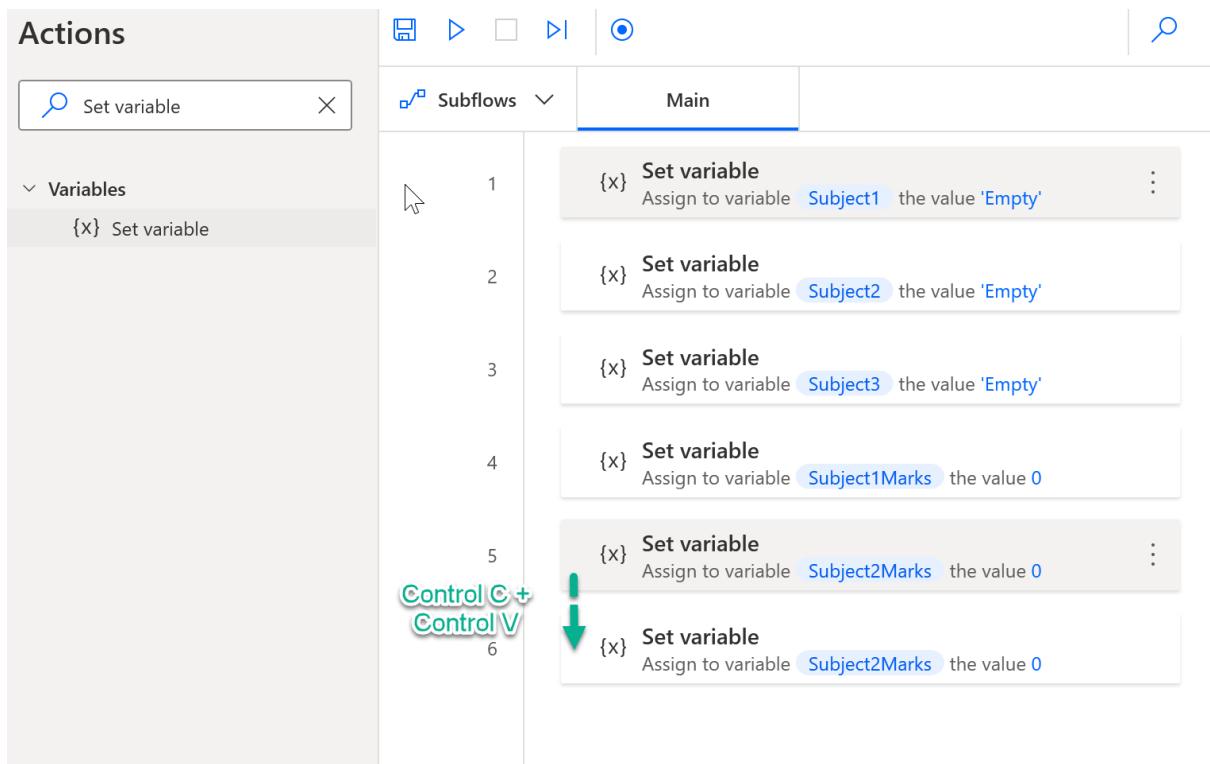
Drag and Drop the Set Variable action and rename it to Subject1Marks with a default value of 0. Click on Save.



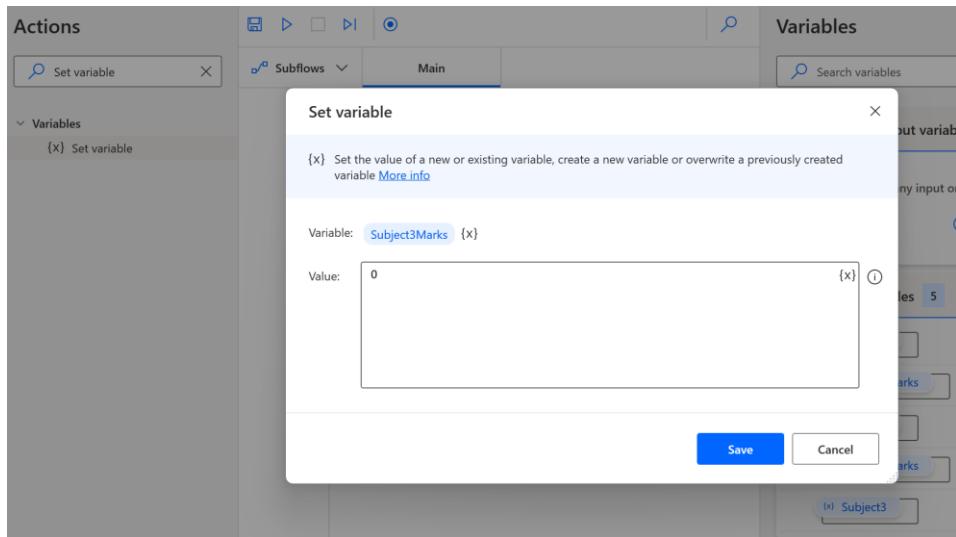
Lets add another variable to hold the Subject2Marks and set it with a default value of 0.Click on Save.



We will add one final variable to save the third subject mark. For this rather than adding the Set Variable action by dragging and dropping, an easier way would be to copy the previous step and paste it which is a supported way to replicate steps in Power Automate Desktop

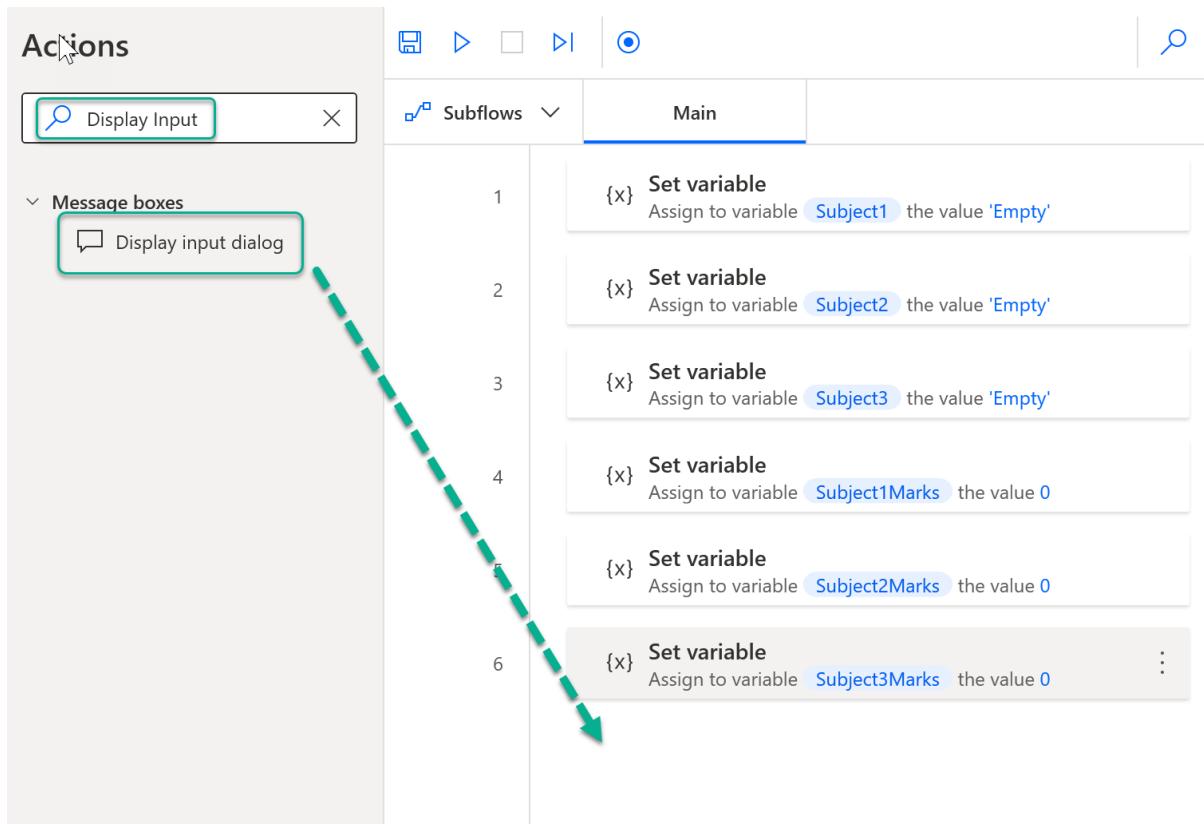


Now click on the step to rename the variable to Subject3Marks and set value to 0 and click on Save.

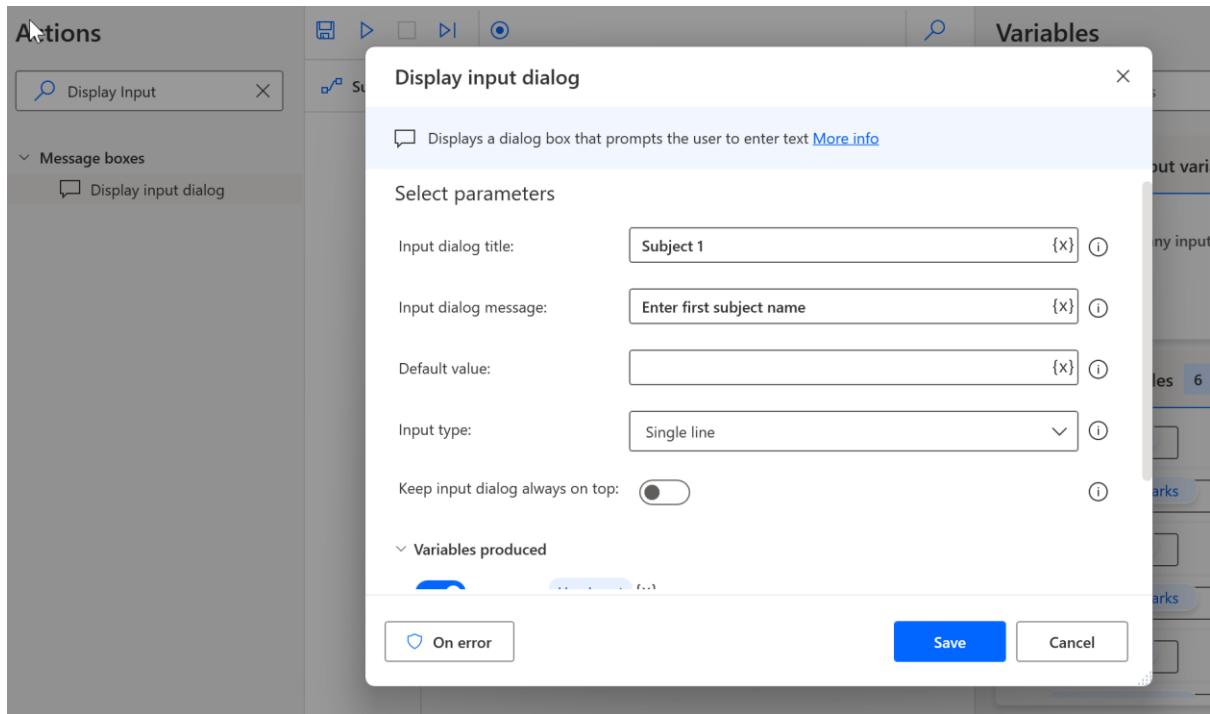


Dynamic Input Collection

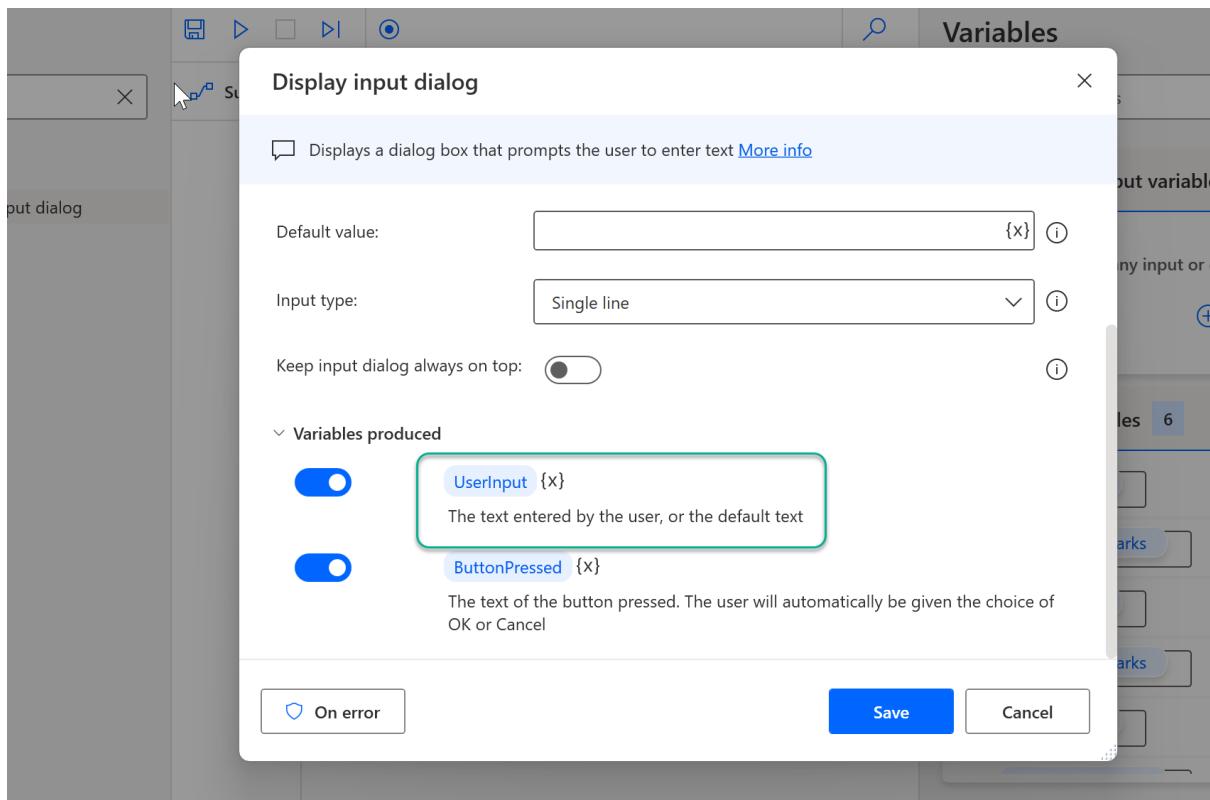
Now we want to dynamically collect the input from users for the Subject names for which we will use the Display input dialog action . Search for it and drag and drop it to the design studio



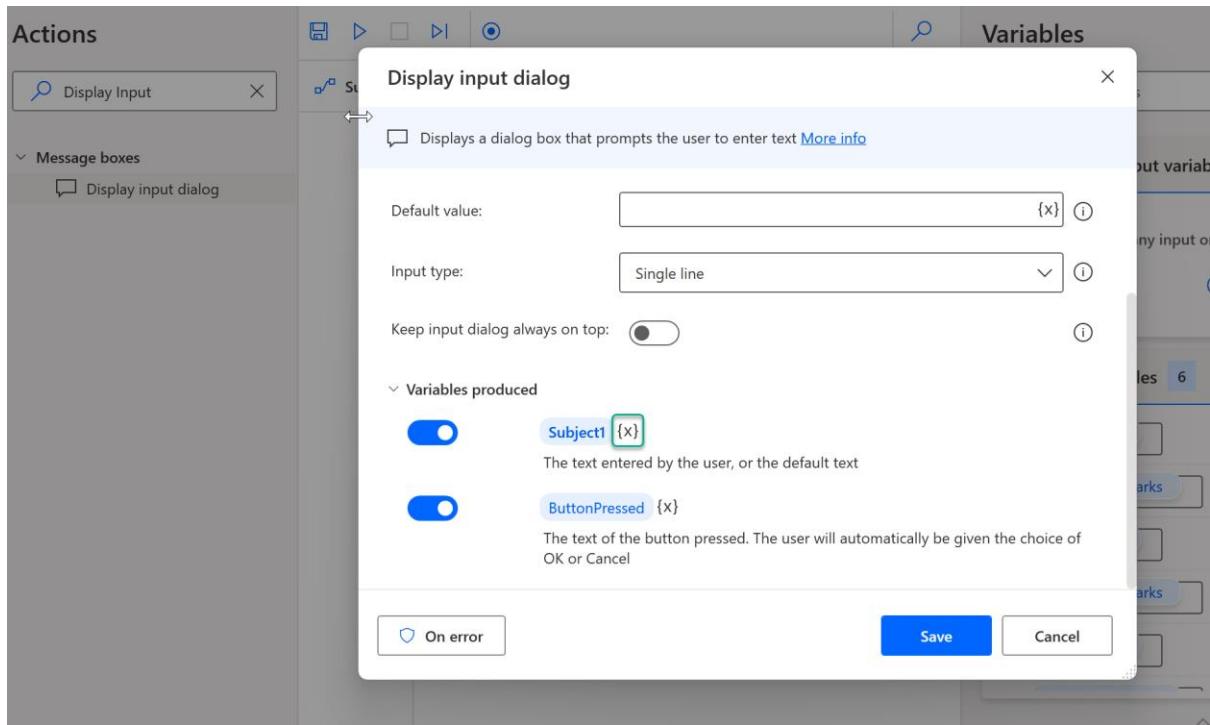
Lets specify the parameters for the input dialog box as below



In the same dialog, we will rename the output variable which will hold the subject name. By design the variable name is UserInput, but as we have already created a variable by the name subject 1, we will replace the variable output with our variable.



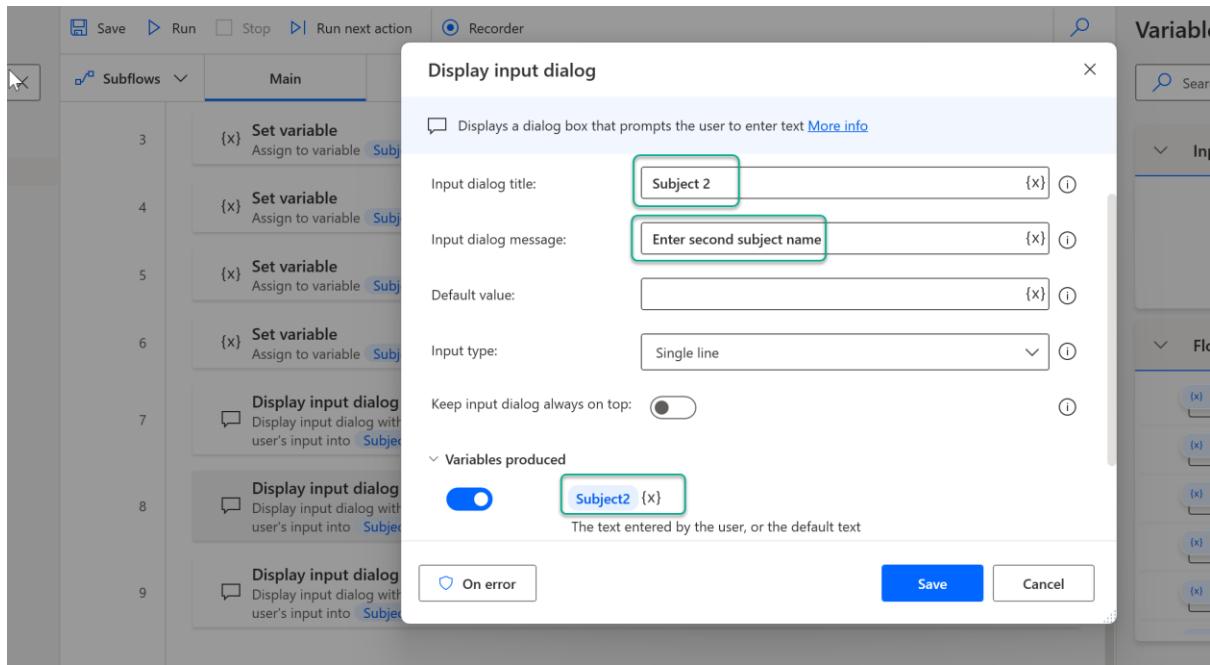
Click on the X symbol and select Subject1 from the variable list. Click on Save



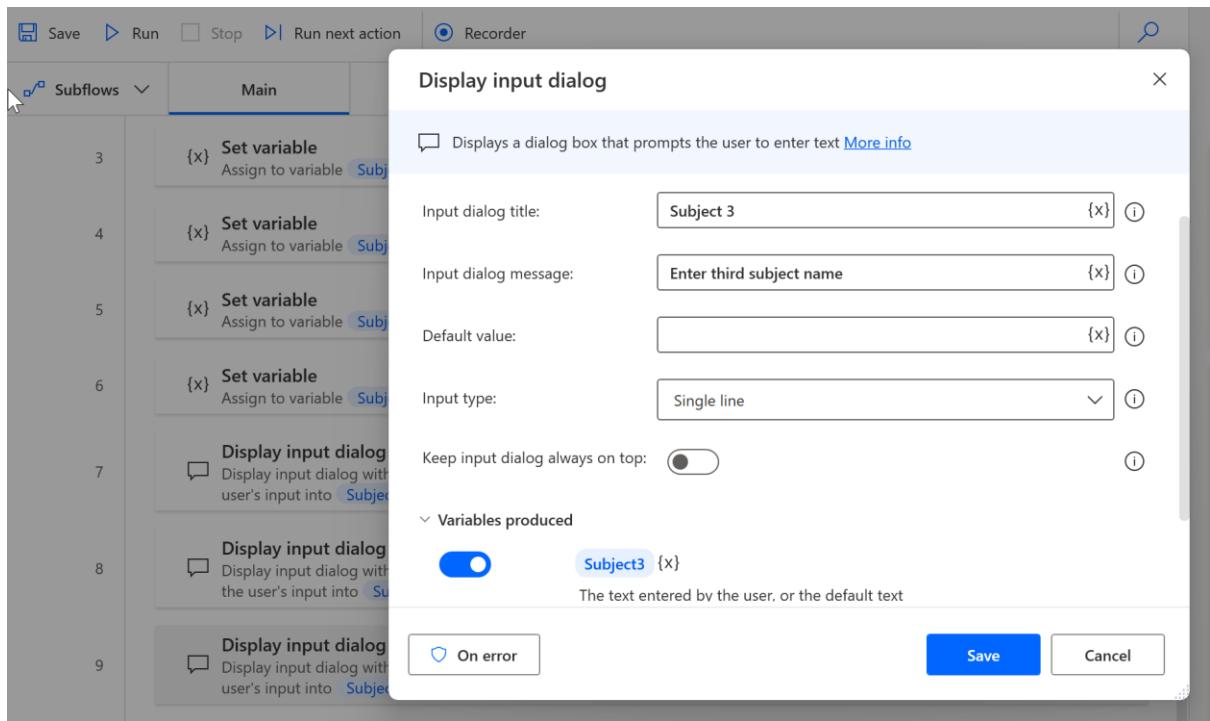
Lets copy and paste this action 2 times as we will repeat the input collection process to get the rest of the 2 subject names

	Main
3	{x} Set variable Assign to variable Subject3 the value 'Empty'
4	{x} Set variable Assign to variable Subject1Marks the value 0
5	{x} Set variable Assign to variable Subject2Marks the value 0
6	{x} Set variable Assign to variable Subject3Marks the value 0
7	Display input dialog Display input dialog with message 'Enter first subject name' in the notification popup window with title 'Subject 1' and store the user's input into Subject1 and button pressed into ButtonPressed
8	Display input dialog Display input dialog with message 'Enter first subject name' in the notification popup window with title 'Subject 1' and store the user's input into Subject1 and button pressed into ButtonPressed
9	Display input dialog Display input dialog with message 'Enter first subject name' in the notification popup window with title 'Subject 1' and store the user's input into Subject1 and button pressed into ButtonPressed

Lets take the first action copy and edit it to capture the second subject details as below and click on Save.

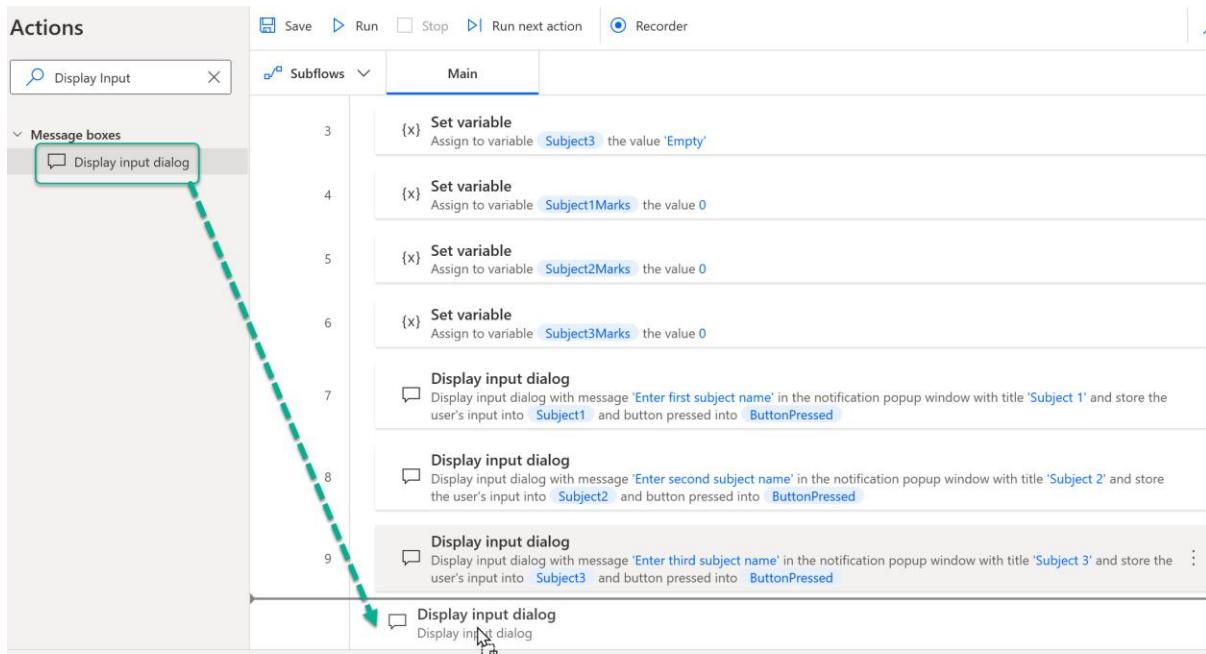


Let update the next action copy to reflect the third subject data capture and click on Save

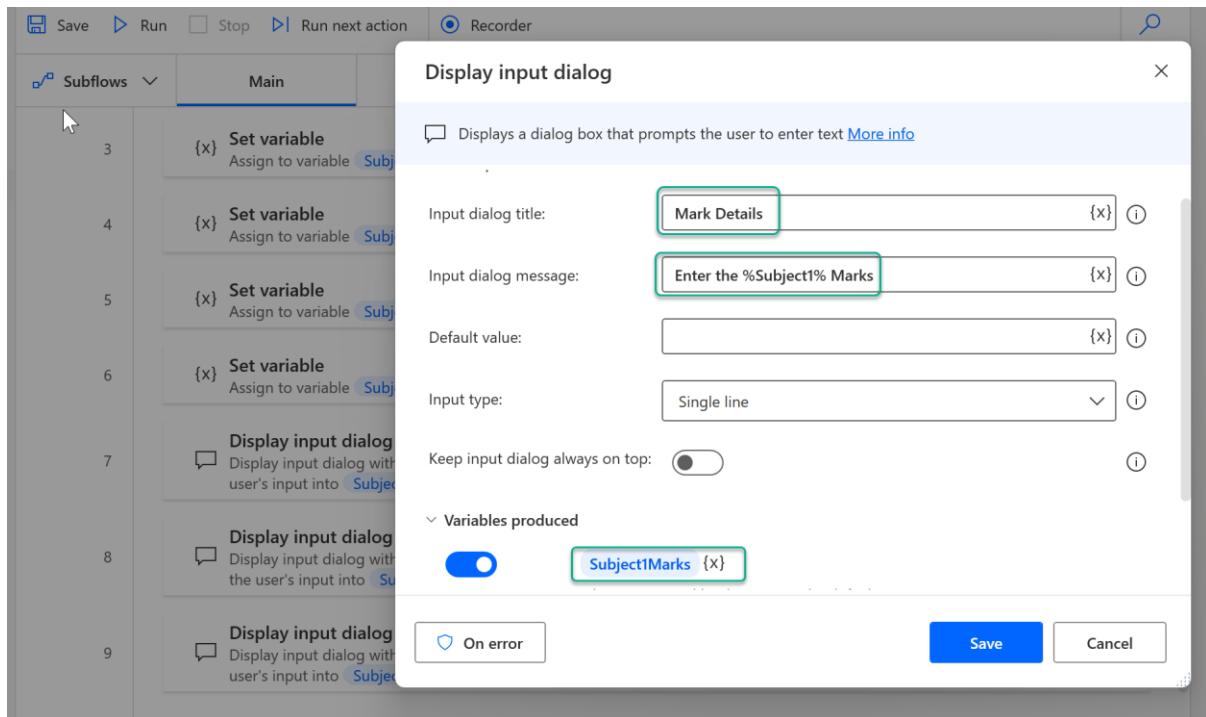


Capture Student Marks

Now lets capture the student marks by dragging and dropping the Display input dialog action to the studio



Make the parameter changes for the input dialog as below and change the output variable of the mark capture to Subject1Marks which we had already created previously. Click on Save.



Let's make a copy of the previous action twice so that we will edit it to capture the marks of the second and third subject

The Actions panel displays a list of 12 'Display input dialog' actions. The first action (7) has its message set to 'Enter first subject name' and stores the user's input into 'Subject1' and 'ButtonPressed'. Subsequent actions (8-12) follow a similar pattern, incrementing the subject number and the variable names. The Variables panel on the right lists flow variables: ButtonPressed, ButtonPressed2, Subject1, Subject1Marks, and Subject2.

Lets edit the first action copy and make the below parameter changes and click on Save

The 'Display input dialog' configuration dialog is open, showing the following settings:

- Input dialog title:** Mark Details
- Input dialog message:** Enter the %Subject2% Marks
- Default value:** (empty)
- Input type:** Single line
- Keep input dialog always on top:** (unchecked)

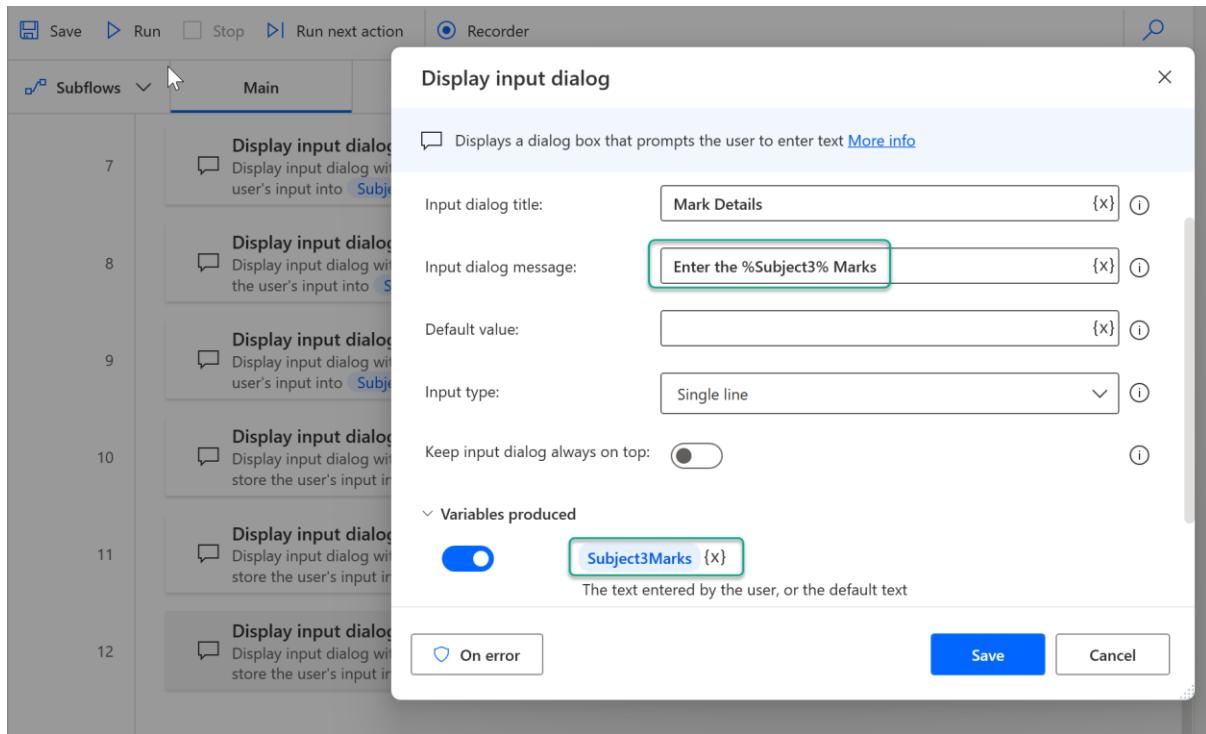
Variables produced:

- Subject2Marks {x} (highlighted with a green border)

Note: The text entered by the user, or the default text

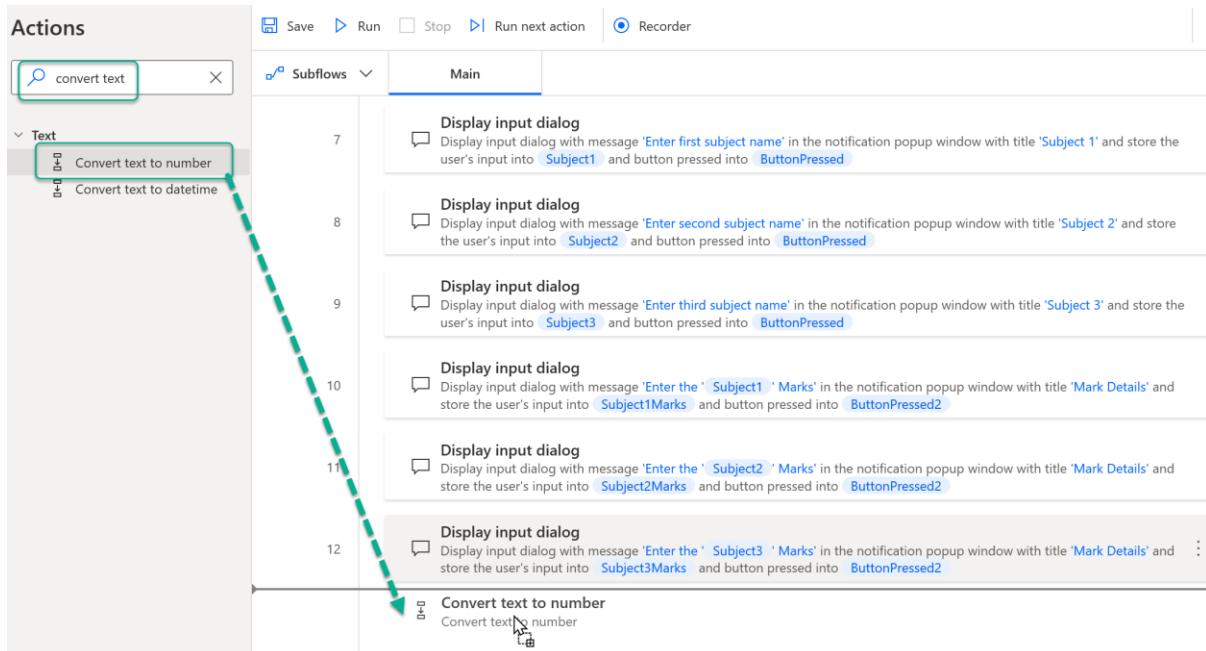
Buttons: On error, Save, Cancel

Lets make the similar change to the next copied action as well and click on Save



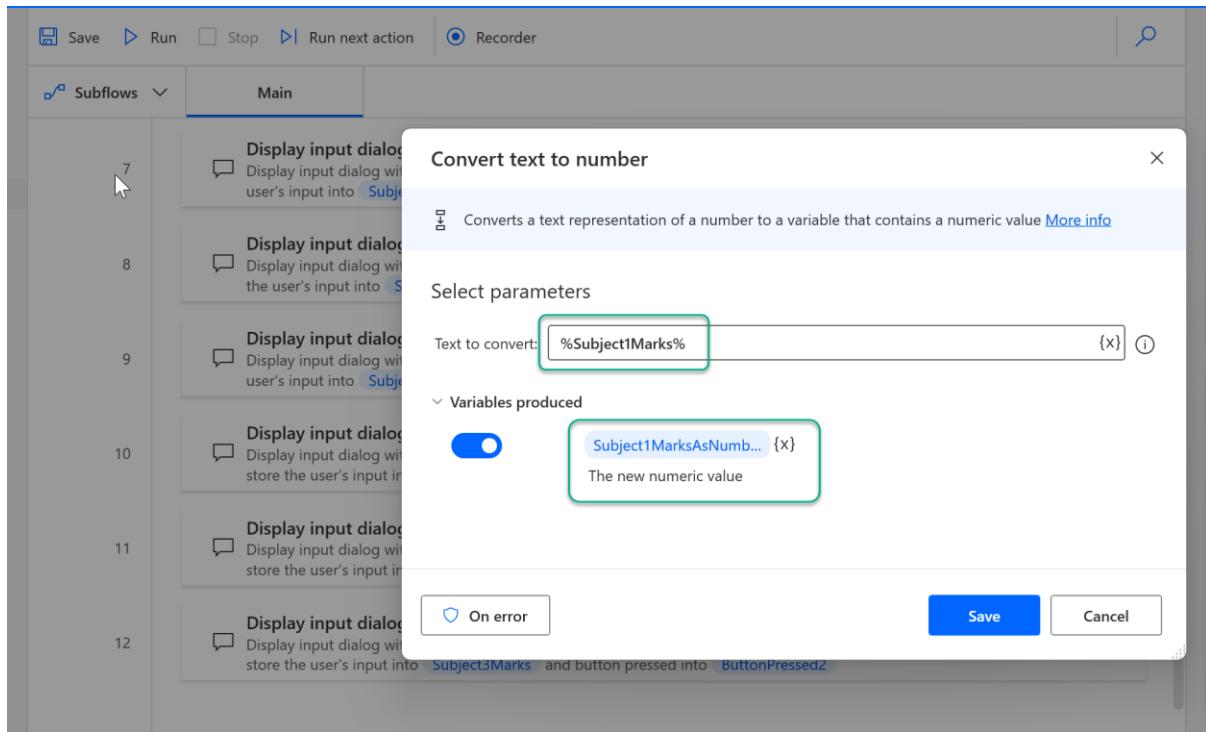
Number variables & Convert Text to Number Action

Though the marks entered may look like numbers they are in fact stored as text. So will use the *Convert text to number* action to convert it to number and store it in a number variable.



Specify the text to convert as the main parameter. In our case it is the Subject Marks. So we will use this action for all 3 subjects. For subject 1 specify the parameters as below and click Save

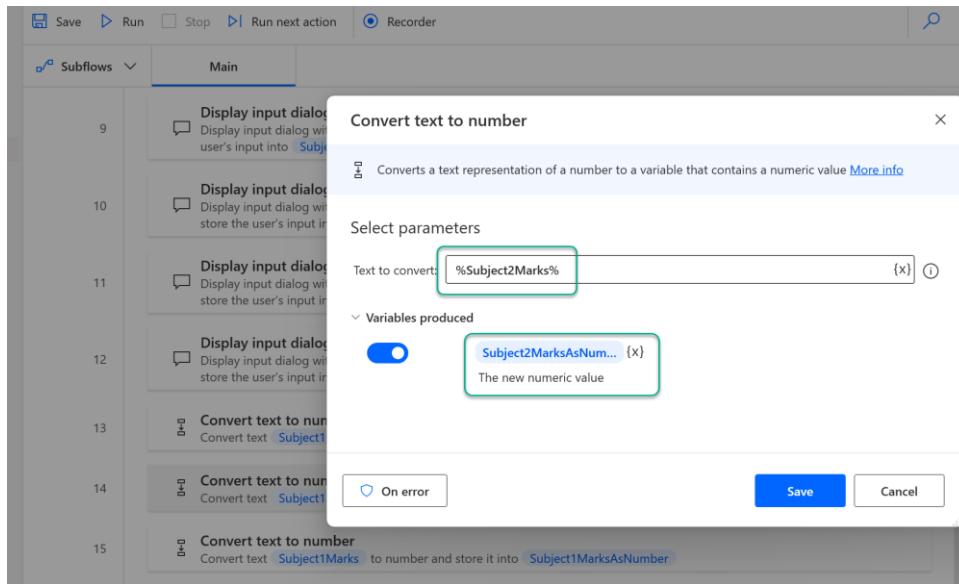
The new number output will be stored in the variable Subject1MarksAsNumber



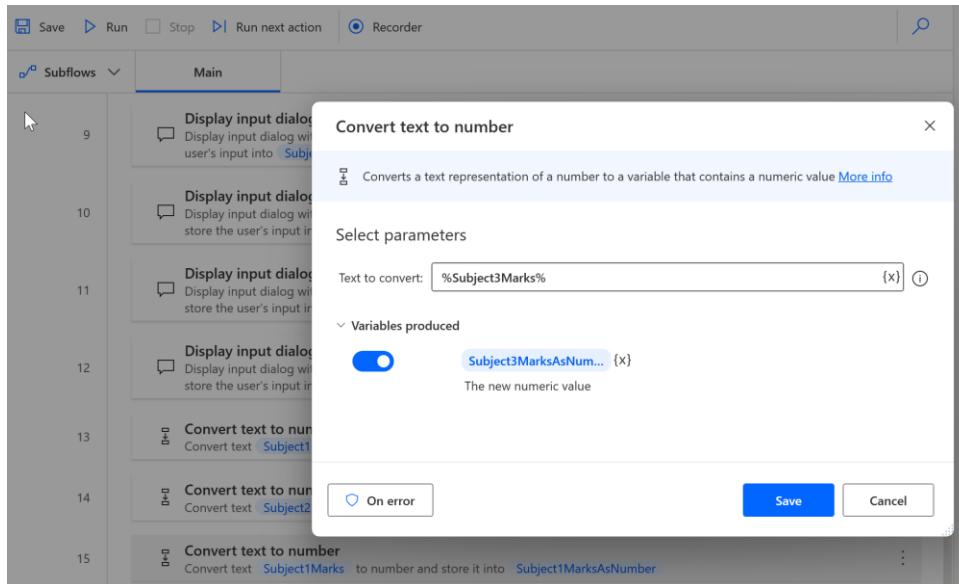
Let's replicate this action 2 times to make changes for rest of the subjects as well

The screenshot shows a software interface with a sidebar titled 'Actions' containing a search bar and a list of actions: 'convert text' under 'Text'. The main window shows a sequence of actions: 'Display input dialog' (9), 'Display input dialog' (10), 'Display input dialog' (11), 'Display input dialog' (12), 'Convert text to number' (13), 'Convert text to number' (14), and 'Convert text to number' (15). The 'Convert text to number' actions are highlighted with a green border.

We will edit the first action copy to reflect Subject 2 mark conversion to number .Click on Save



Lets make the change to the next action copy as well and click on Save



Thus we have completed the mark conversion to respective number format

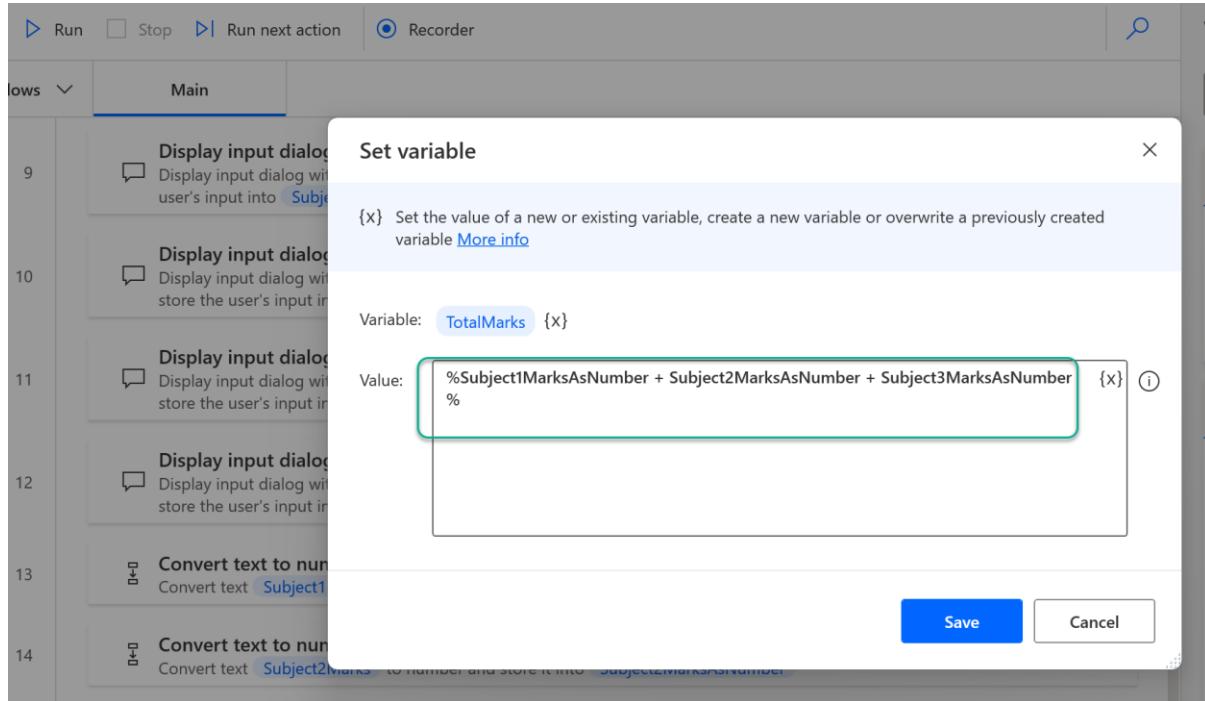
Perform Variable Calculation

Now that we have the needed variables in place, we will see how to add the 3 subject marks. Lets add a set variable action to the design studio . Specify the variable name as TotalMarks. Though it is not created previously, it will be auto created as part of this action.

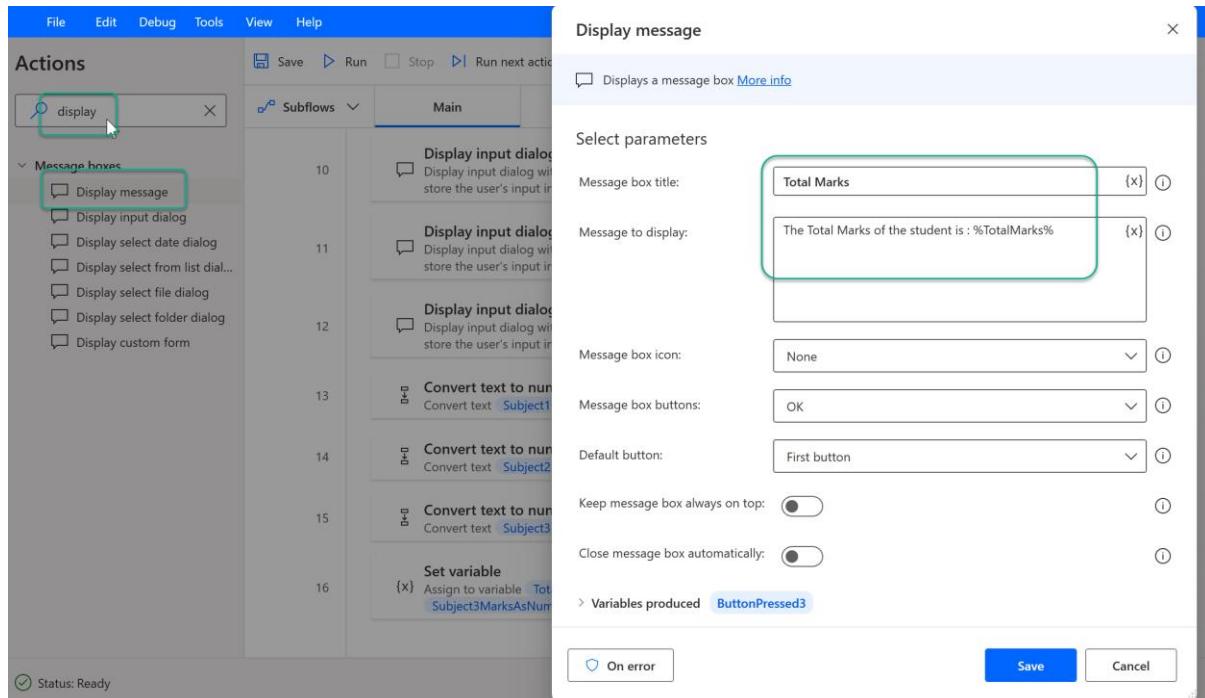
In the value field, Select the X sign and add the Subject1MarksAsNumber, Subject2MarksAsNumber, Subject3MarksAsNumber variables and add Plus sign in between them . However this would result in the variable values being concatenated. So we need to avoid that.

To do that, let's make a change by encapsulating all the variable names in one single percentage leading and trailing format as below. This will result in the variable values to be added and outcome will be placed in the Total Marks variable.

Click on Save



Lets display the total marks using Display Message action and add the Total Marks variable in the *Message to display* field. Click on Save

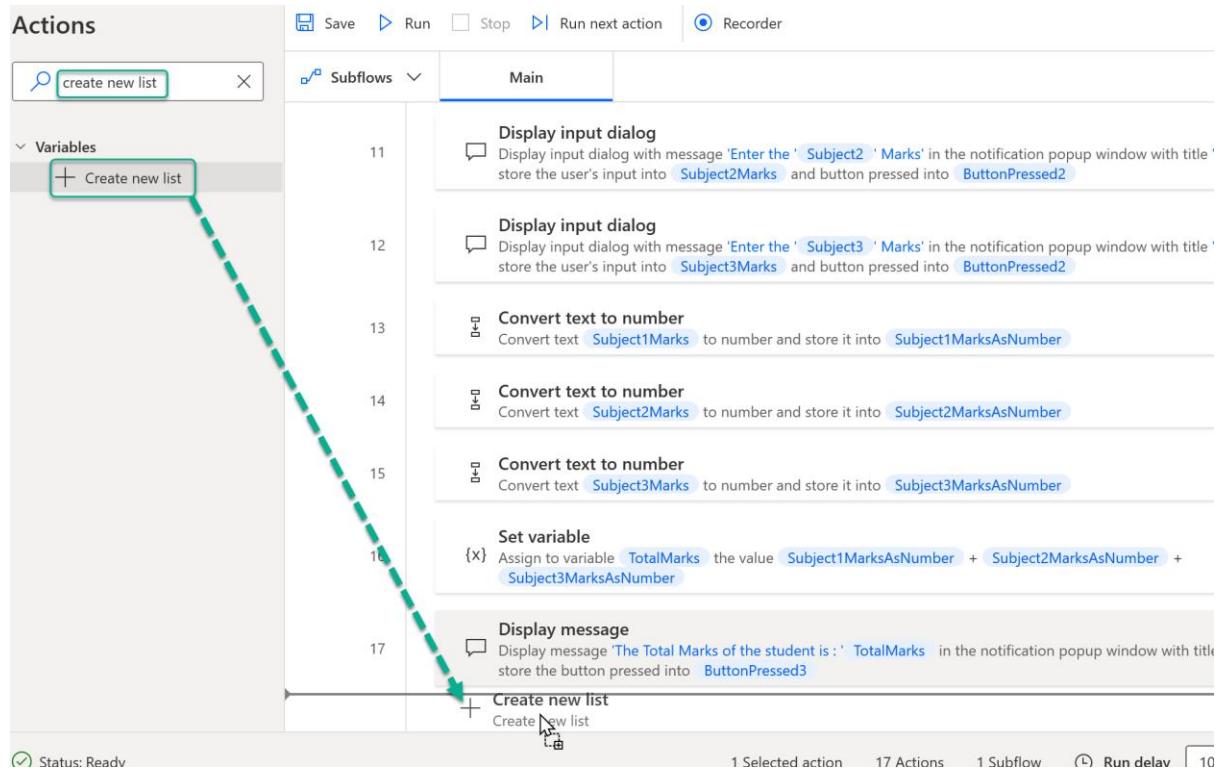


List Variable

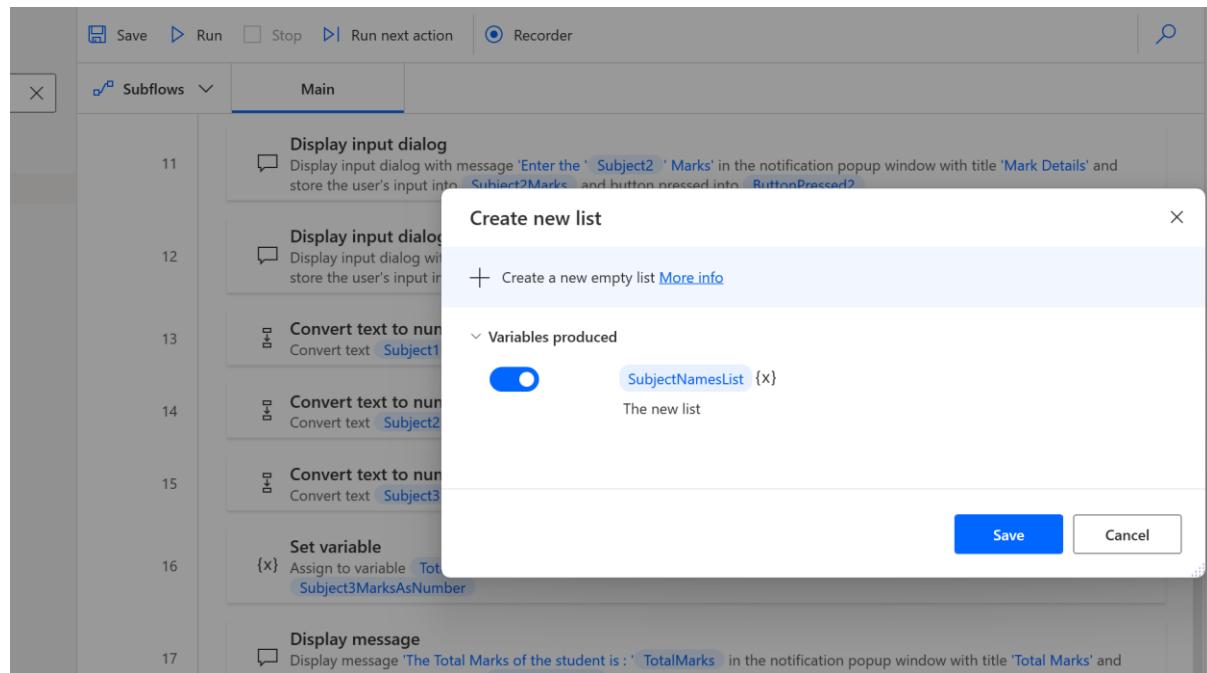
List variables are used when we need to store more than a single value. For instance we want to store the 3 subject names together in a variable, we can use the List variable. We will see the

advanced list operations later, but for now, we will see how to create the list and store the values in the list variable.

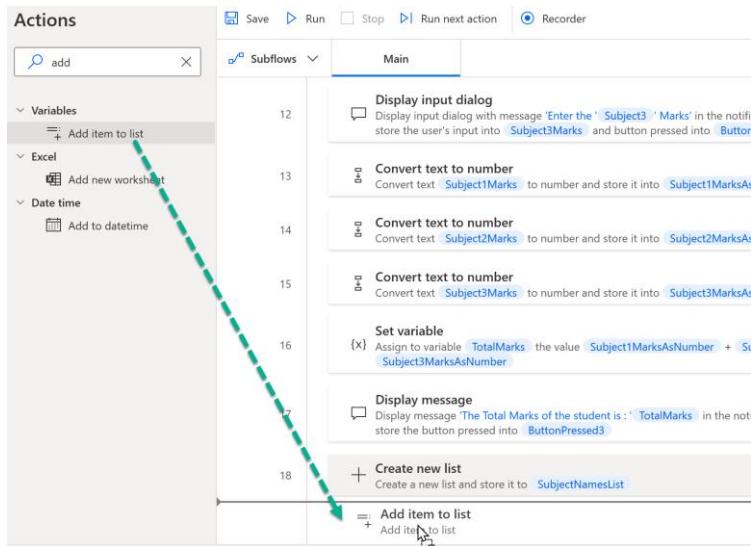
To add a new list, drag and drop the Create new List action to the studio



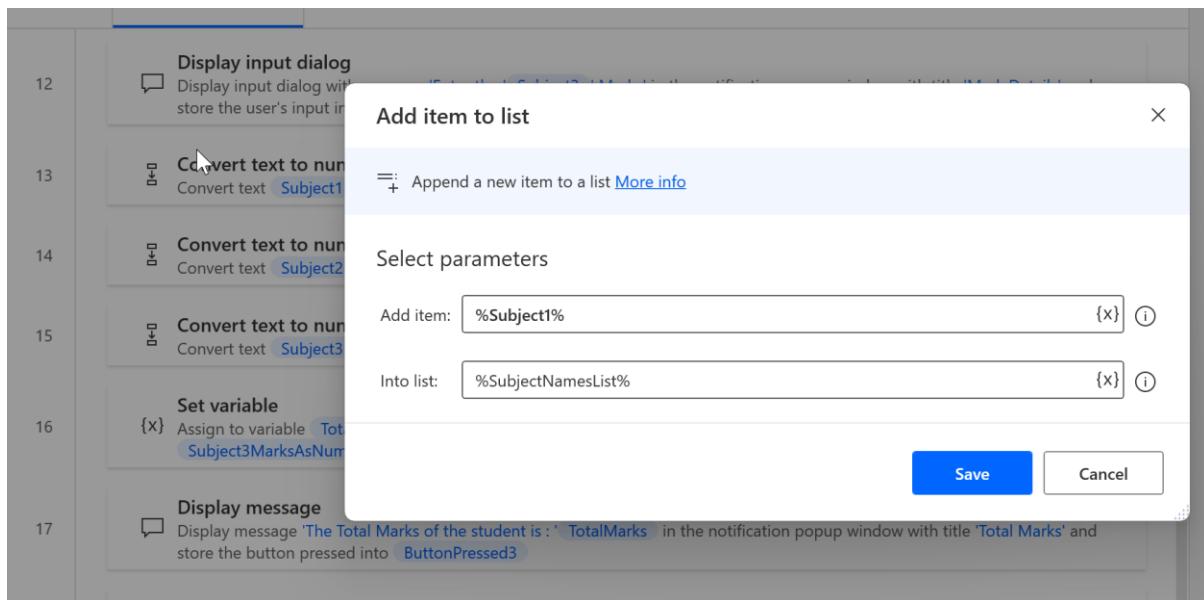
The new list can then be configured with a name as below.



We can add items to the list using the *Add item to list action*

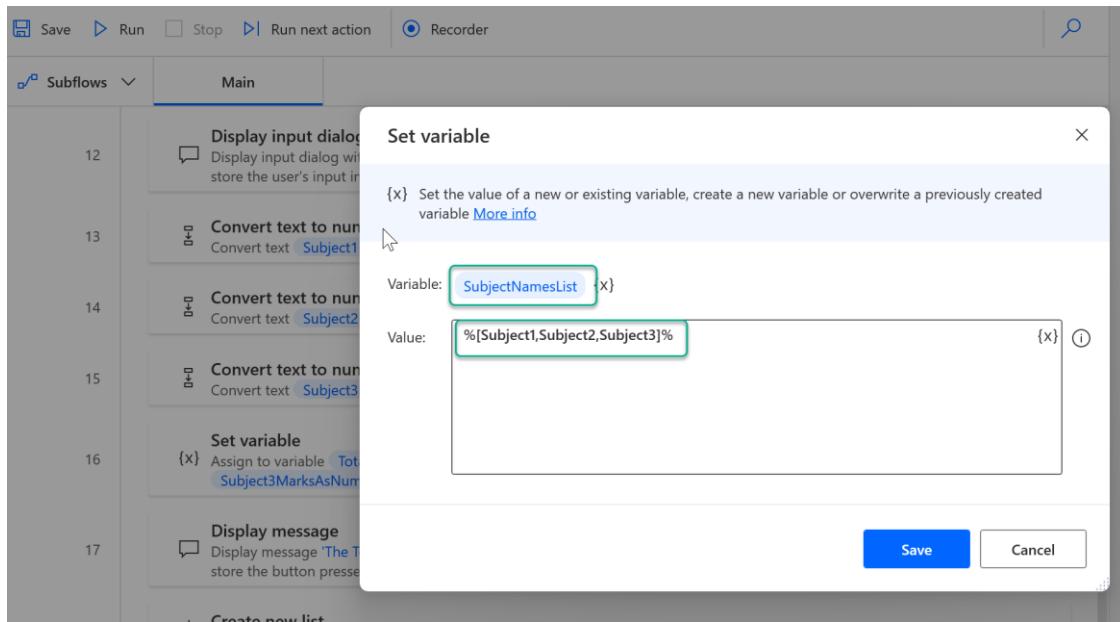


But this allows us to add one items at a time



So as an easy hack, we will not use the Add item to list action, rather we will rely on the Set variable action to add multiples items at a time. So lets add the Set variable action to the design studio and we will add the 3 subject names by specifying their variables names within % symbols as below surrounded by Square brackets. Each variable will be separated by a comma.

Click on Save



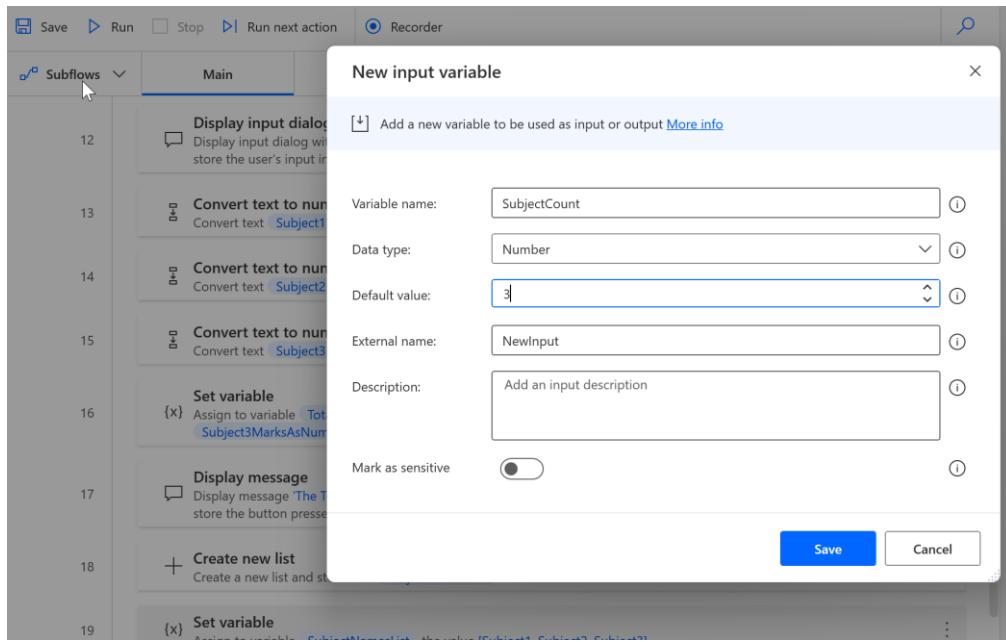
Input/Output Variable

Before we head over to the average calculation, we will see the usage of Input/Output variables. This comes in handy when you want to predeclare some variables before the flow run or to pass values to other flows. In this case we have 3 subjects and we will store the count of subjects in an Input Variable as this is going to be static through out the run and not going to change.

To do that , head over to the Input/Output variable section and declare a variable *SubjectCount* and set it to 3

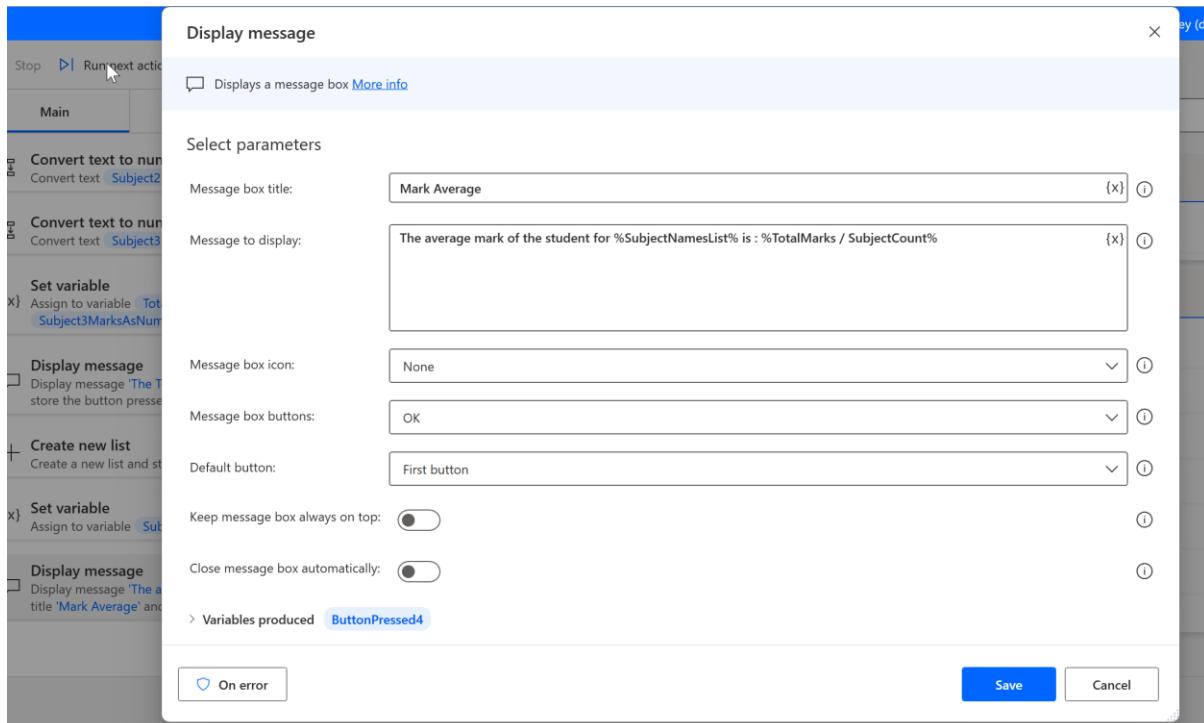
The screenshot shows the 'Variables' section of the flow editor. An 'Input / output variables' section is highlighted, showing a placeholder message: 'There aren't any input or output variables here yet'. Below this, there are two buttons: '1' (with a plus sign) and '2' (with an input icon). The 'Flow variables' section lists several variables: ButtonPressed1, ButtonPressed2, ButtonPressed3, Subject1, Subject1Marks, and SubjectNamesList.

We will specify the Input variable name and choose the Number data type and give the default value as 3. Click on Save.



Mark Average Calculation

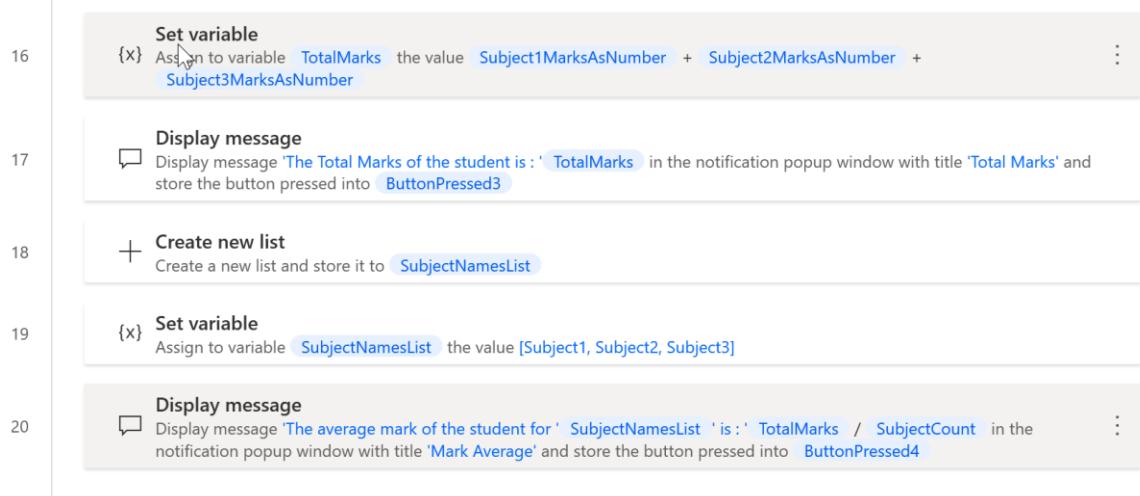
Now we will calculate the average of the marks by using the formula in the Display Message action . So drag and drop the Display message action to the studio and update the formula in the Message to display field as below. *Total Marks/SubjectCount* will give us the average of the student's mark. We are also displaying the subject names which we have saved in the *SubjectNamesList* in the message box . Click on Save



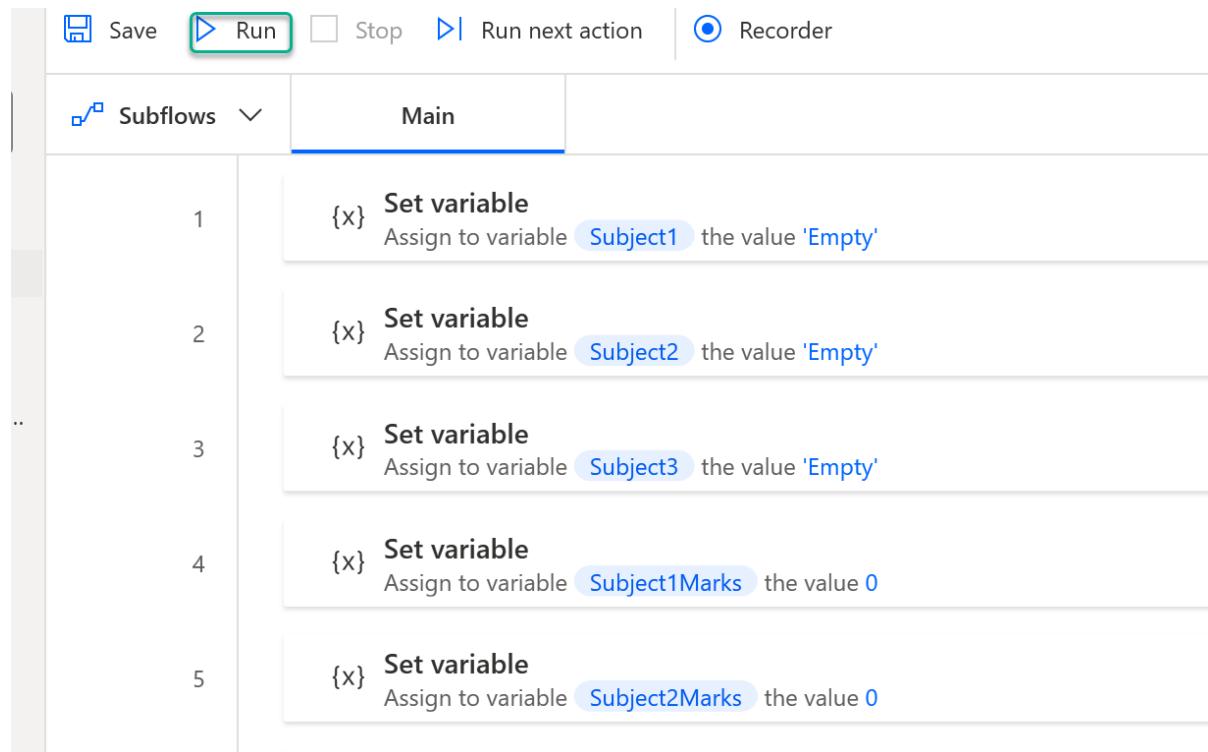
Run the Flow

Thus we have overall 20 actions listed as below :

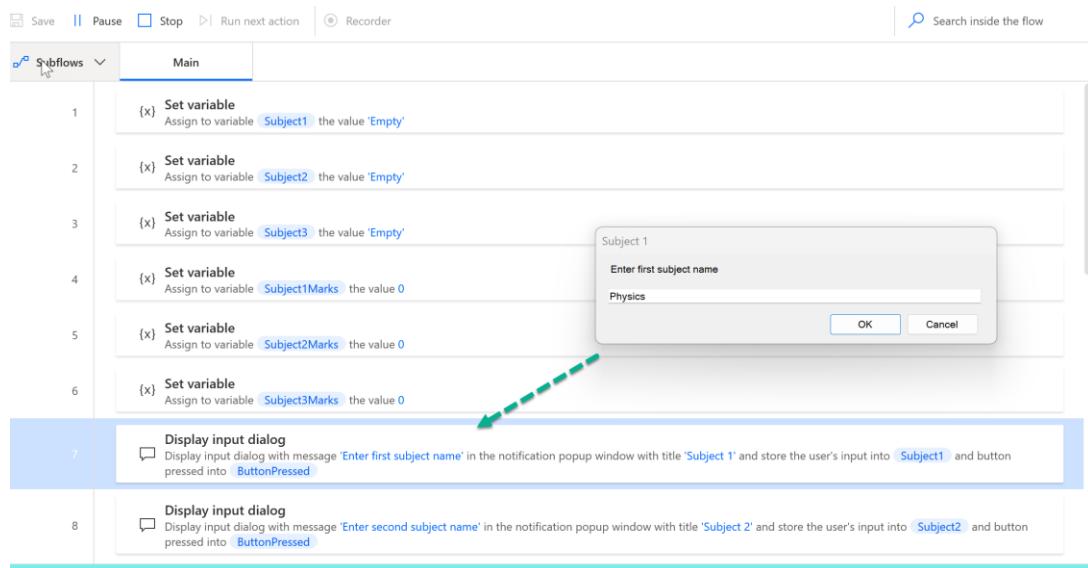
	Main
1	{x} Set variable Assign to variable Subject1 the value 'Empty'
2	{x} Set variable Assign to variable Subject2 the value 'Empty' ⋮
3	{x} Set variable Assign to variable Subject3 the value 'Empty'
4	{x} Set variable Assign to variable Subject1Marks the value 0
5	{x} Set variable Assign to variable Subject2Marks the value 0
6	{x} Set variable Assign to variable Subject3Marks the value 0
7	Display input dialog Display input dialog with message 'Enter first subject name' in the notification popup window with title 'Subject 1' and store the user's input into Subject1 and button pressed into ButtonPressed
8	Display input dialog Display input dialog with message 'Enter second subject name' in the notification popup window with title 'Subject 2' and store the user's input into Subject2 and button pressed into ButtonPressed
9	Display input dialog Display input dialog with message 'Enter third subject name' in the notification popup window with title 'Subject 3' and store the user's input into Subject3 and button pressed into ButtonPressed
10	Display input dialog Display input dialog with message 'Enter the ' Subject1 ' Marks' in the notification popup window with title 'Mark Details' and store the user's input into Subject1Marks and button pressed into ButtonPressed2 ⋮
11	Display input dialog Display input dialog with message 'Enter the ' Subject2 ' Marks' in the notification popup window with title 'Mark Details' and store the user's input into Subject2Marks and button pressed into ButtonPressed2
12	Display input dialog Display input dialog with message 'Enter the ' Subject3 ' Marks' in the notification popup window with title 'Mark Details' and store the user's input into Subject3Marks and button pressed into ButtonPressed2
13	Convert text to number Convert text Subject1Marks to number and store it into Subject1MarksAsNumber
14	Convert text to number Convert text Subject2Marks to number and store it into Subject2MarksAsNumber
15	Convert text to number Convert text Subject3Marks to number and store it into Subject3MarksAsNumber



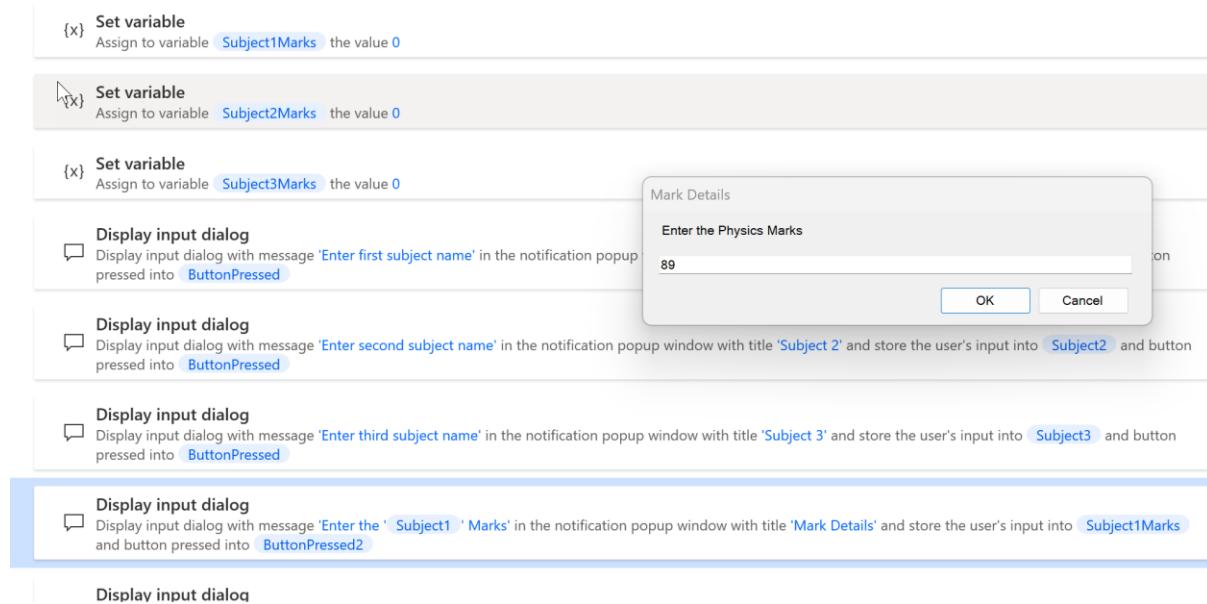
Now that we have created the flow, lets run it by clicking on the Run button at the top



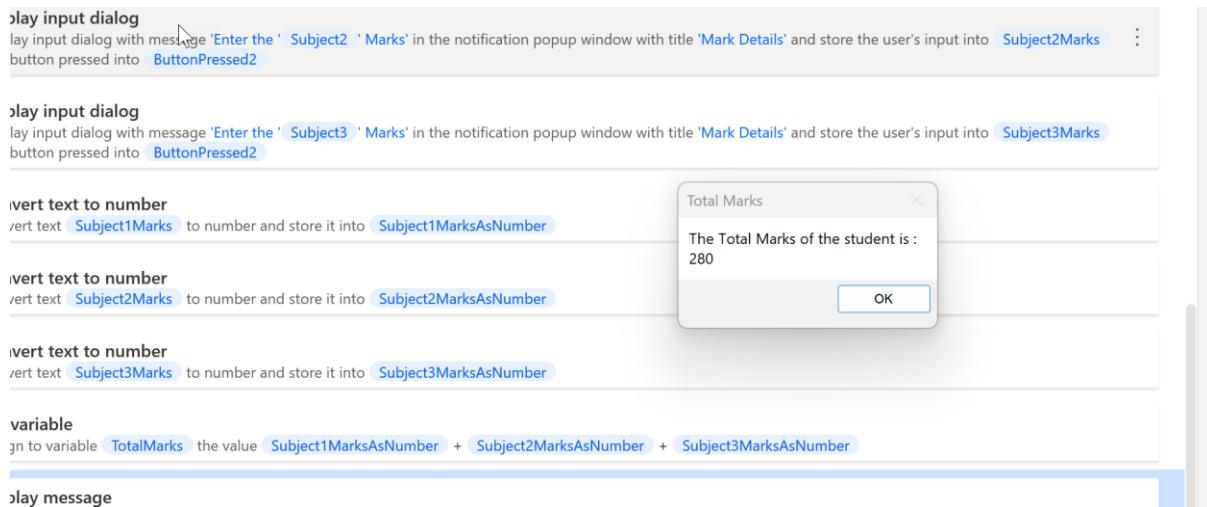
The flow has started running and it has asked the first input which is the subject name , the blue coloured highlight shows which action is being executed.



Similarly it will ask for the rest of the subject names and it will ask the marks for each subject



Finally it will give us the Sum total of the 3 subject's marks as below



It will also calculate the average marks and display it in the dialog message box

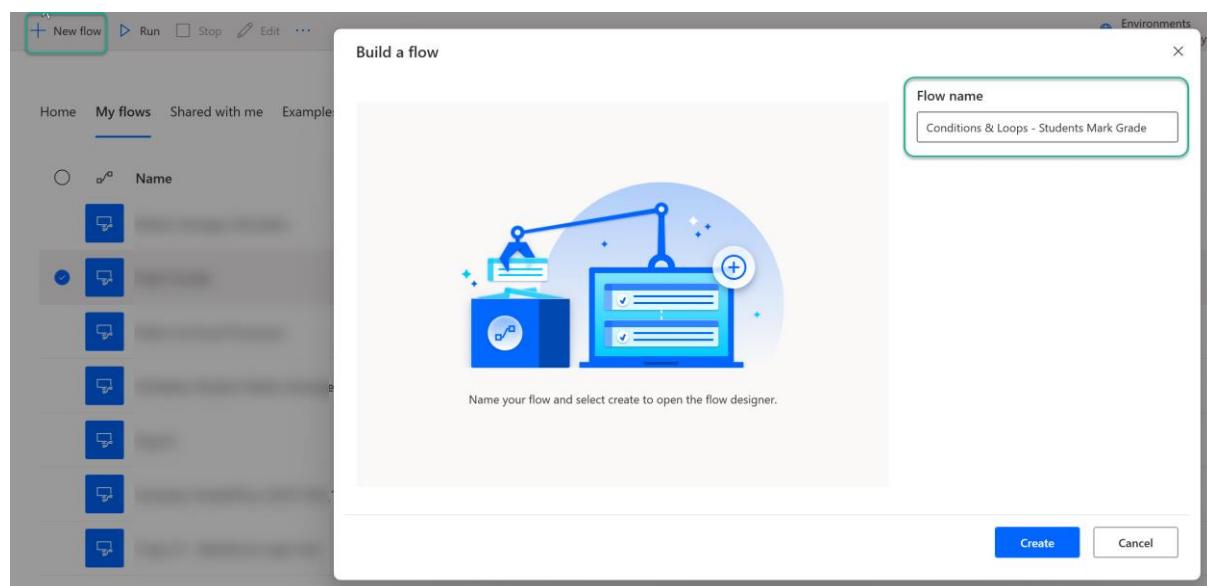
Thus we saw how to make use of various variables like Text, Numbers, List, Input/Output variables to perform the Student's Mark total and average calculations as part of creating this bot.

PAD Exercise 2 – If , Else If and Loops

In this exercise, we will see how we can use the Conditional constructs like If Else as well as explore the looping structures available for use.

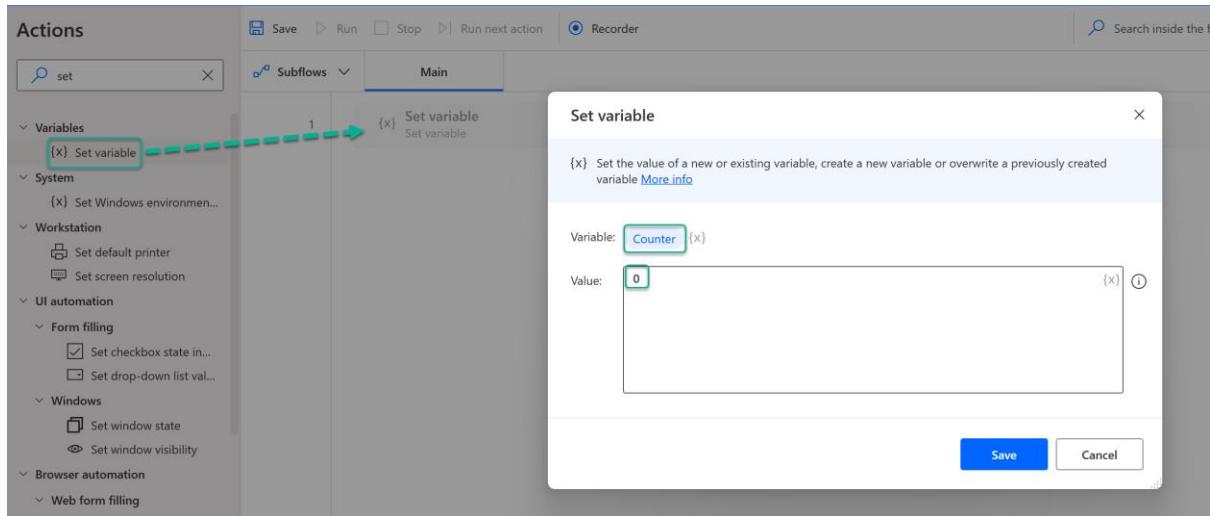
The use case that we would try to implement is to create a bot that can identify the grade of a student based on the total marks he has scored in the exam. We will design the bot in such a way that the bot can run multiple times in a single run to provide the grade information for multiple students.

To get started, let's select *New flow* and specify the flow name. Then click on *Create*.

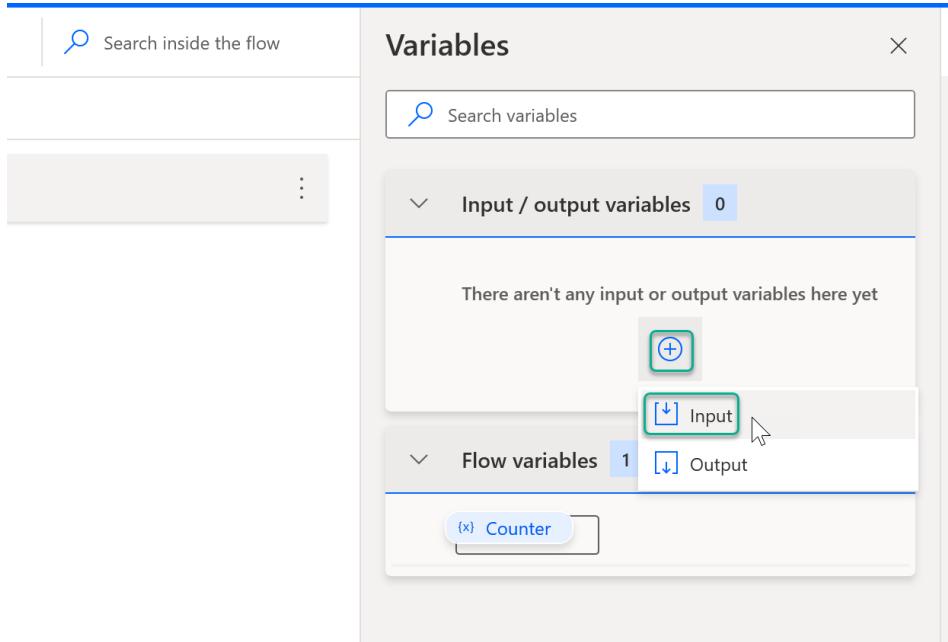


We will add the Set Variable action and add the variable name as Counter to check the Loop exit condition and give it the default value as 0

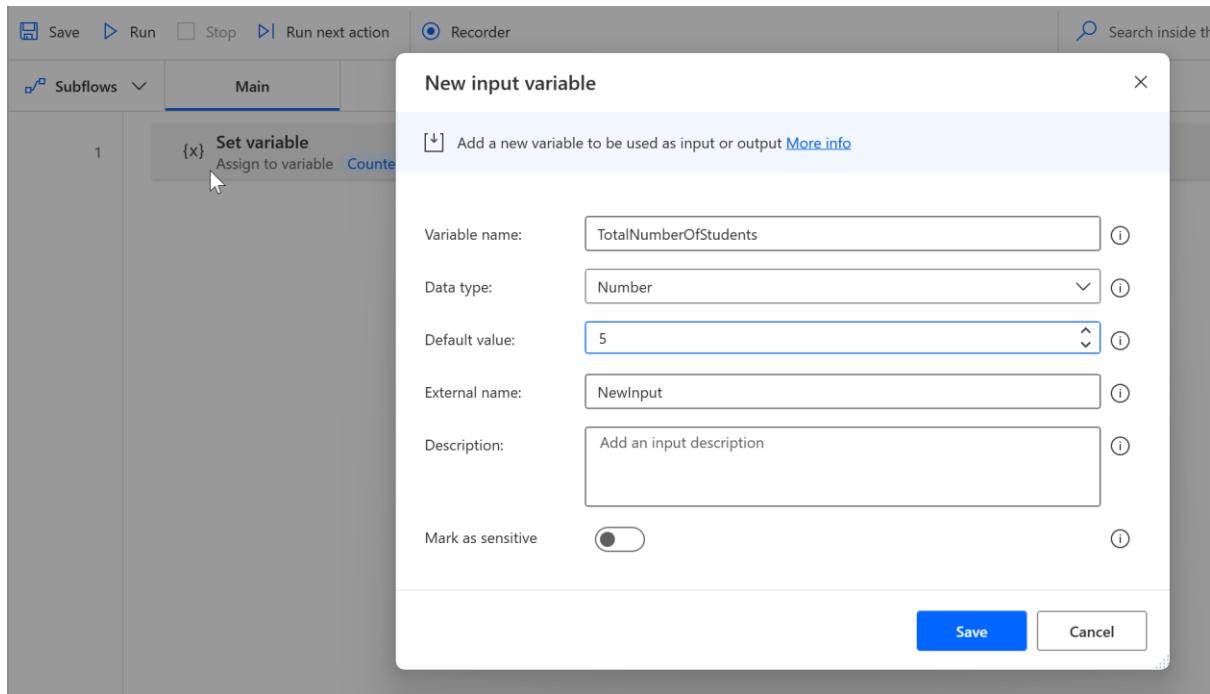
Click on Save



Lets add an Input variable for the Total number of students for which we will be using the grading bot . based on the number of students, we will create a loop to be executed for that many number of times. To add this static variable, lets make use of Input/Output variable option and create an input variable

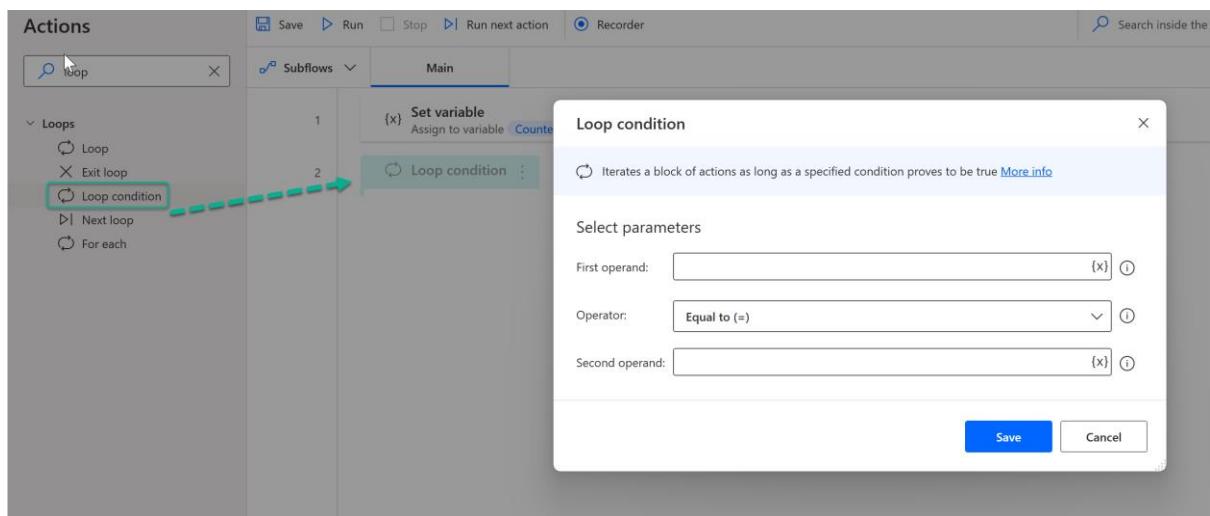


We will specify the variable name and set the value as 5 so that the loop can be controlled to be run for 5 times

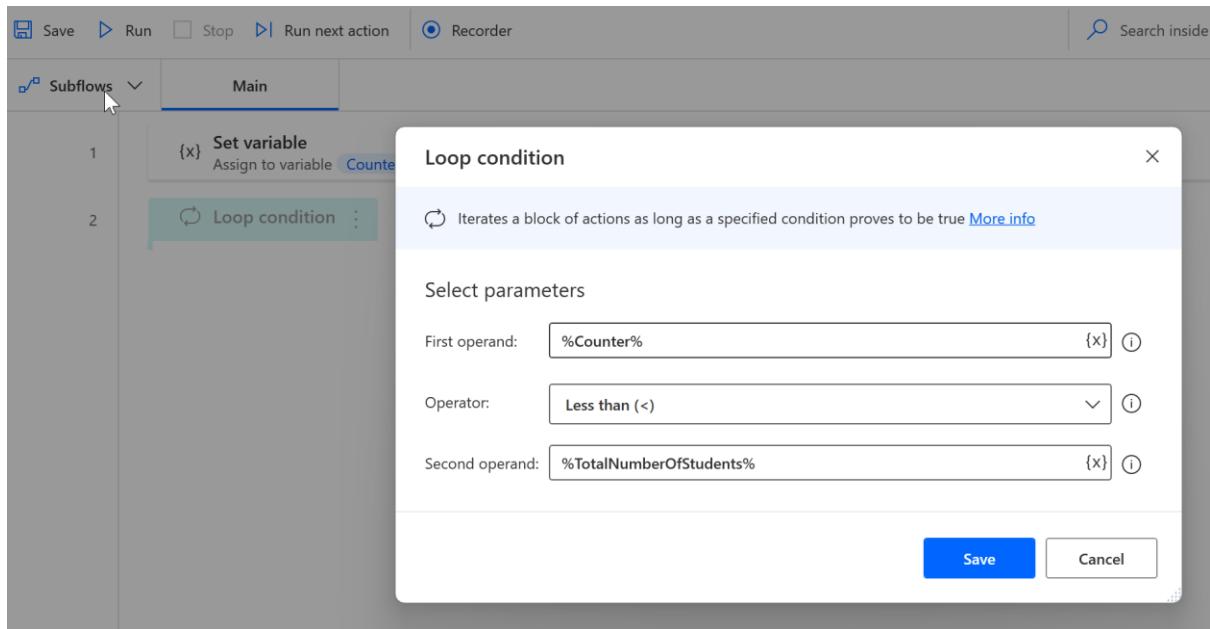


Loop Condition

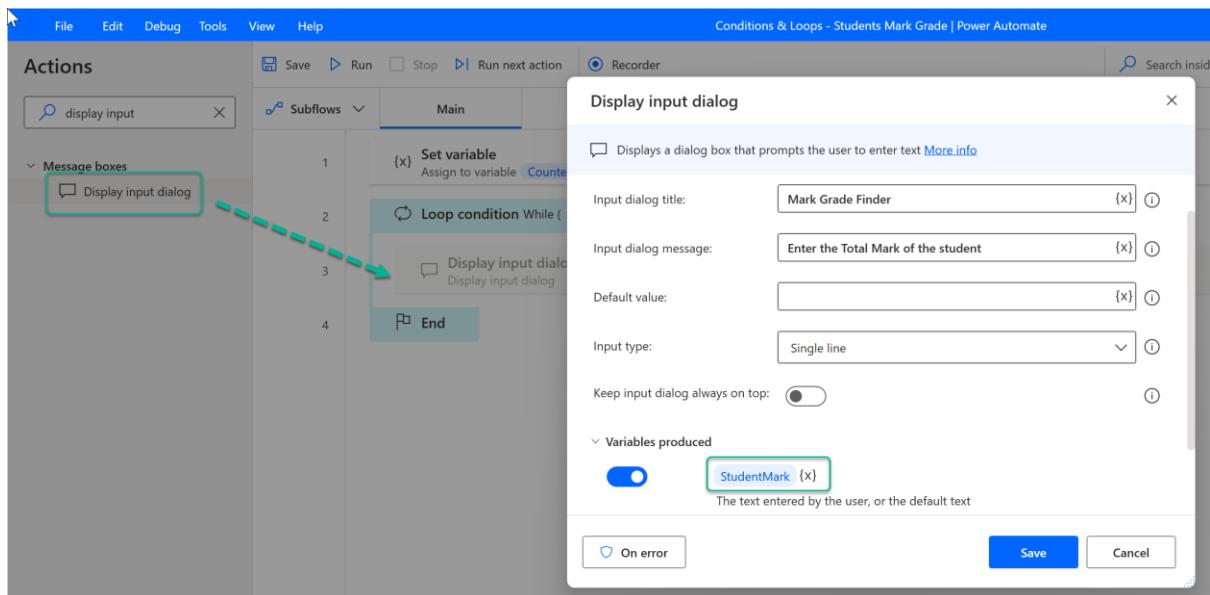
Now, let's add the *Loop condition* action to the design studio. Click on Save.



The loop will run as long as the below condition evaluates to true. Here we are looping for each student and the loop counter will be incremented at the end of each iteration. This way using the below condition, we will iterate till the counter value is less than the *TotalNumberOfStudents* variable value

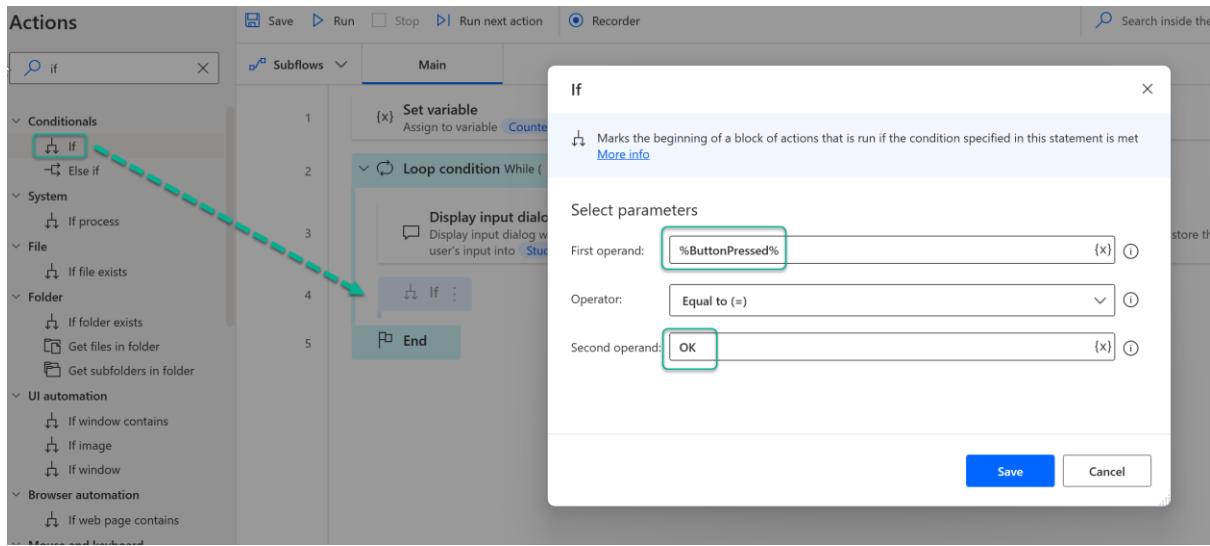


So as to accept the mark of the student for grade analysis, we will add the Display Input dialog action inside the loop . Specify the dialog title and the dialog message and we will rename the output variable as StudentMark.

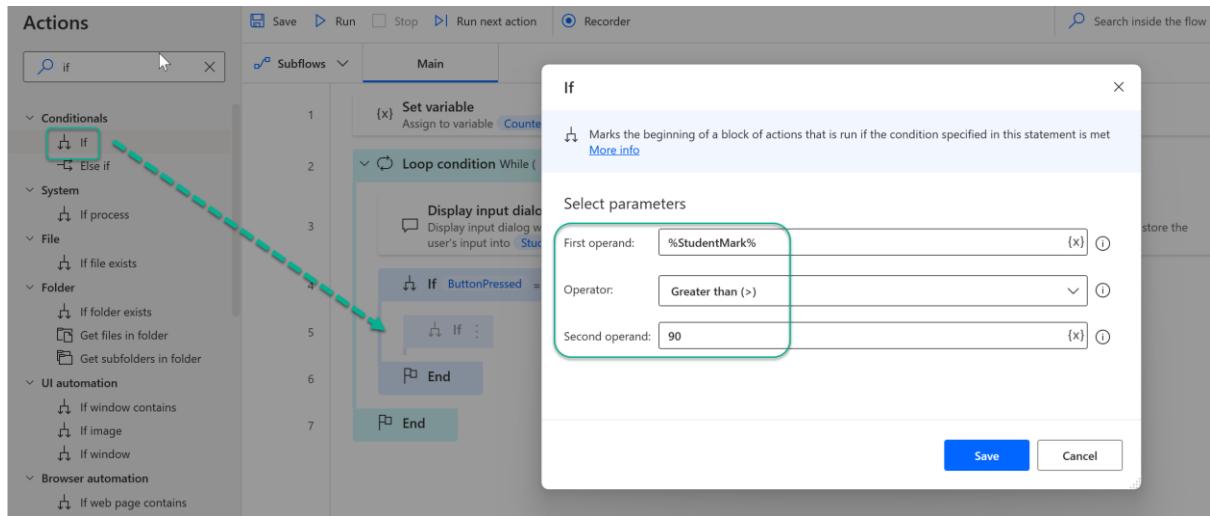


If - Else If – Else Conditions

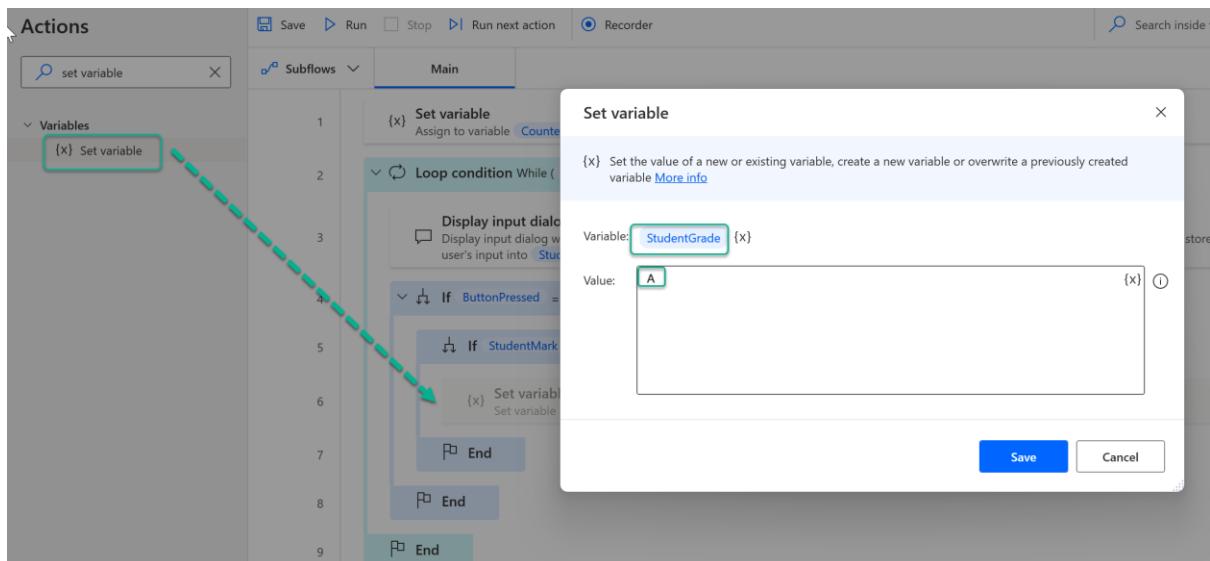
When the Input Dialog comes up the user can either enter the data and click on OK or Cancel. We need to ensure that the flow proceeds only if the OK button is pressed. This value will be present in the Button Pressed output variable of the Input dialog. So as the next step, lets check if the user has indeed pressed the OK button before we do any grade analysis



Within this If, we will add the further logic so that it executes only if the user has pressed the OK button of the initial input dialog. Let's add the nested If to check the mark value and add the StudentMark variable and check if it is greater than 90 so that we can assign the 'A' grade

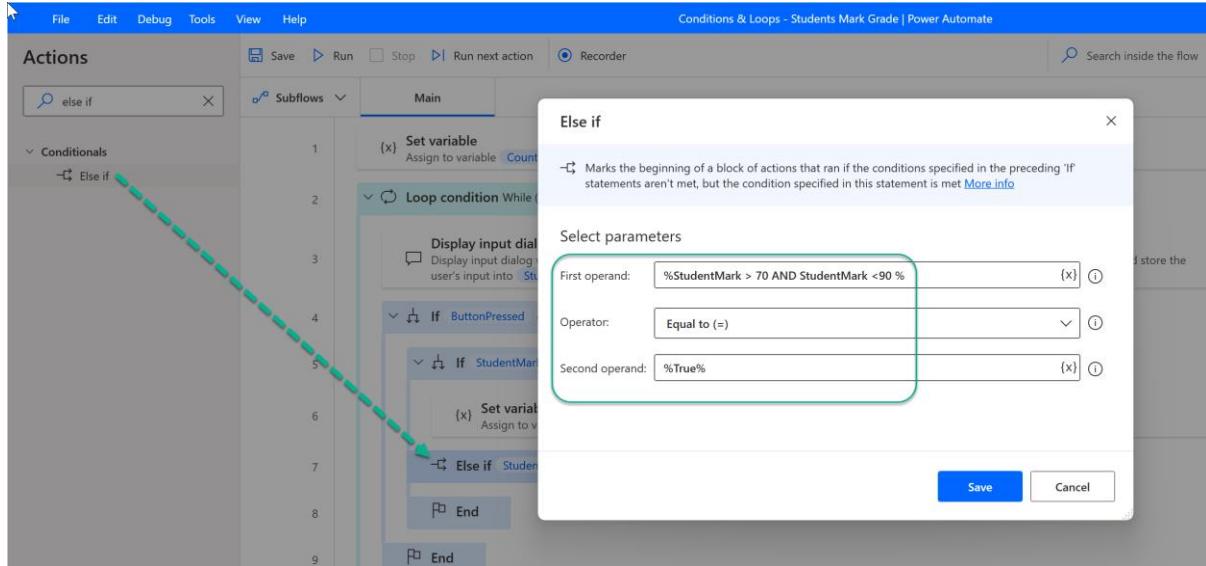


Lets use the Set Variable action to declare a new variable *StudentGrade* and set it to value A

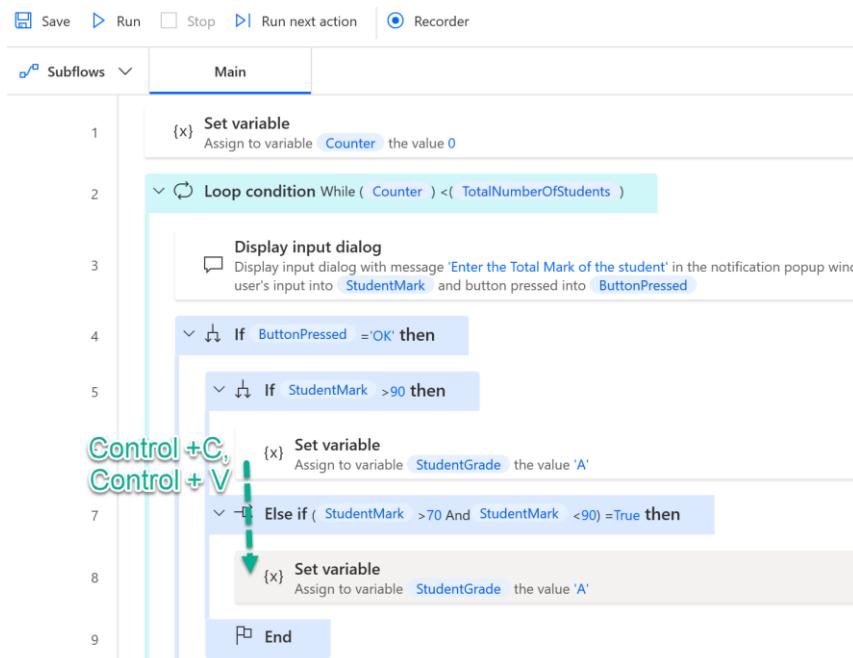


Boolean Variable

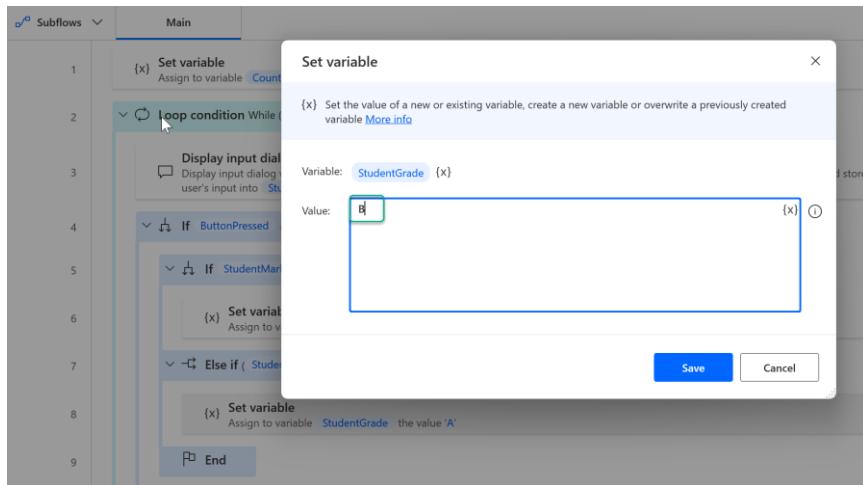
As the next step add an else if condition inside the nested if . We will check 2 conditions to decide the B grade. Student's Mark has to greater than 70 and less than 90. Since Else if block has the option to evaluate just 2 operands, we will slightly take a different approach here and in the First Operand we will check for both the condition using an AND operator so that it will evaluate to a True only if both are True. Then we will evaluate it against the Boolean variable which is represented as %True% to give the final logical output of the Else If comparison.



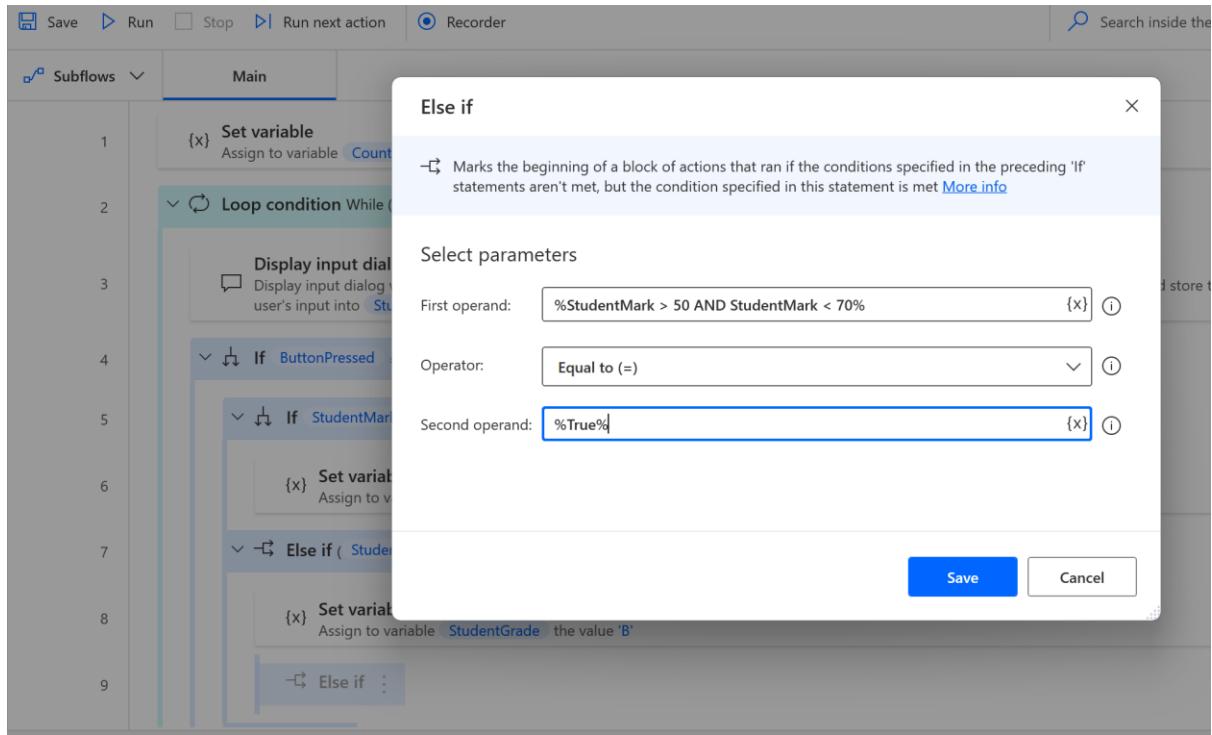
Now we need to set the Grade value to the Grade Variable for which we can copy the previous variable setting action to the else if block



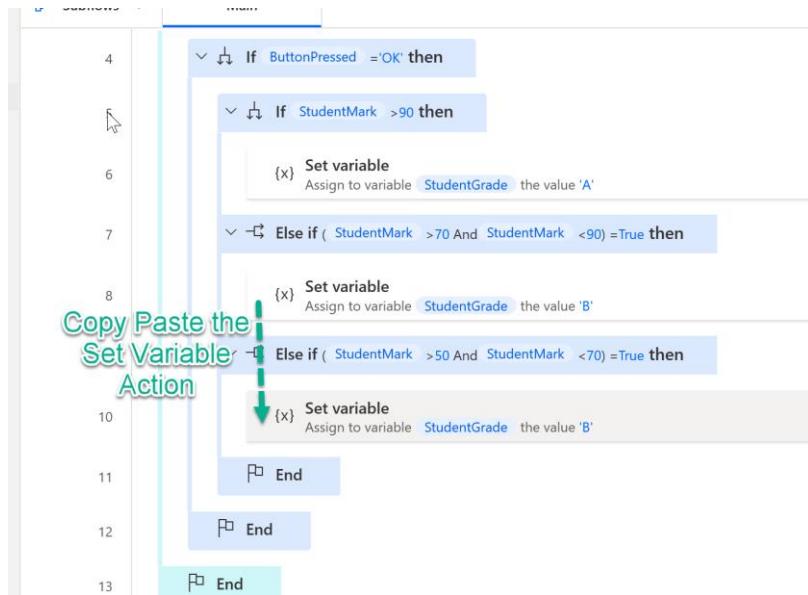
Let click on the copied action to edit the parameters



Lets add another Else If action to the design studio just below the previous Else If action. This time we need to check if the mark is greater than 50 and Less than 70 and evaluate it to True



Lets copy paste the set variable action once again to assign the Grade C to the variable



Click on the copied action to edit the parameters

Save Run Stop Run next action Recorder Search inside the f

Subflows Main

```

4   ↘ If ButtonPressed
5     ↘ If StudentMark
6       {x} Set variable
         Assign to variable StudentGrade {x}
7     ↘ - Else if ( StudentMark > 70 And StudentMark < 90 ) = True then
8       {x} Set variable
         Assign to variable StudentGrade the value 'B'
9     ↘ - Else if ( StudentMark > 50 And StudentMark < 70 ) = True then
10      {x} Set variable
         Assign to variable StudentGrade the value 'B'
11    ↗ End
12  ↗ End
13 ↗ End

```

Set variable

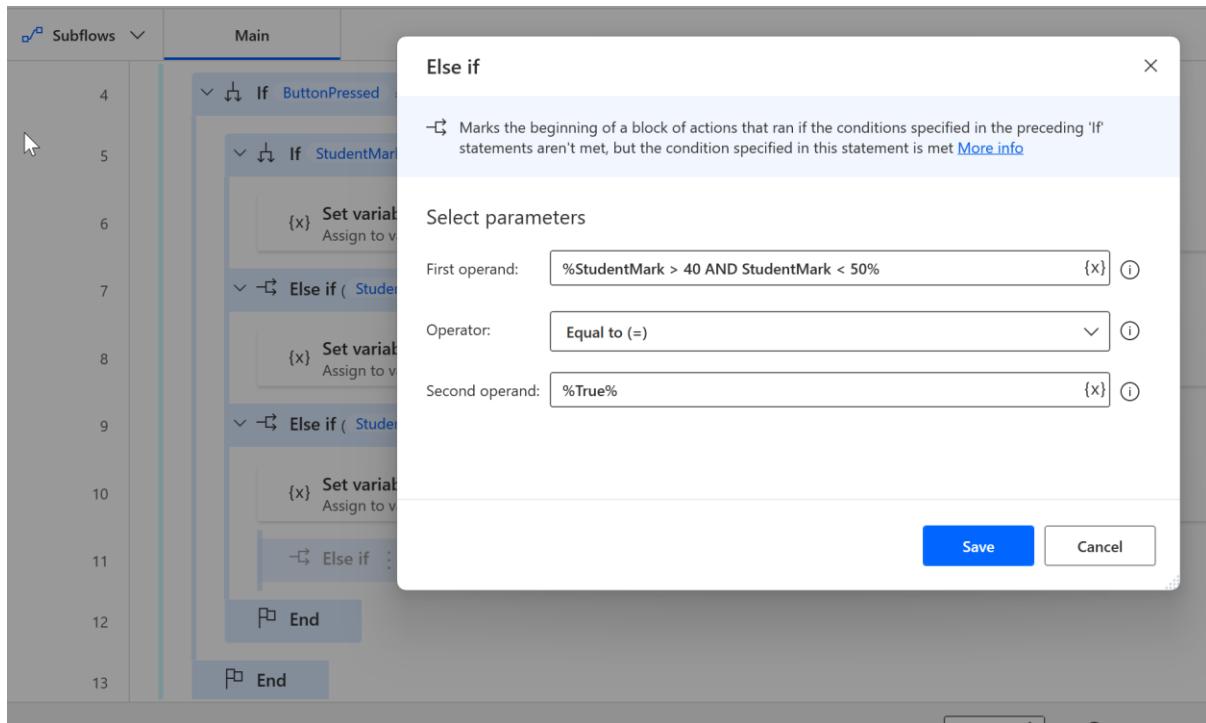
{x} Set the value of a new or existing variable, create a new variable or overwrite a previously created variable [More info](#)

Variable: `StudentGrade` {x}

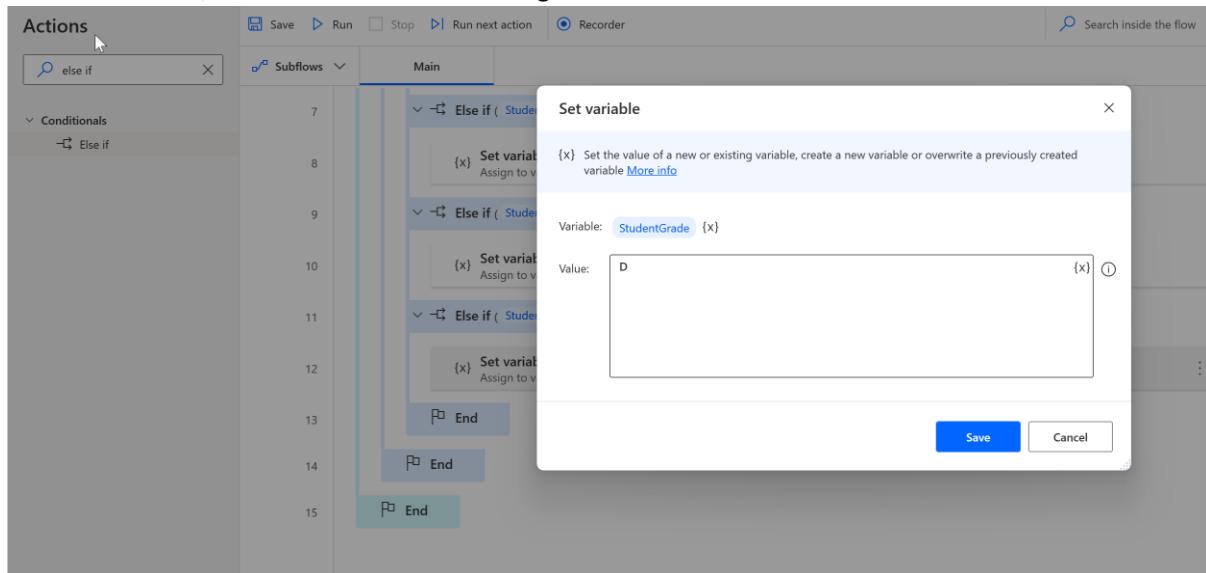
Value: C {x} ⓘ

Save Cancel

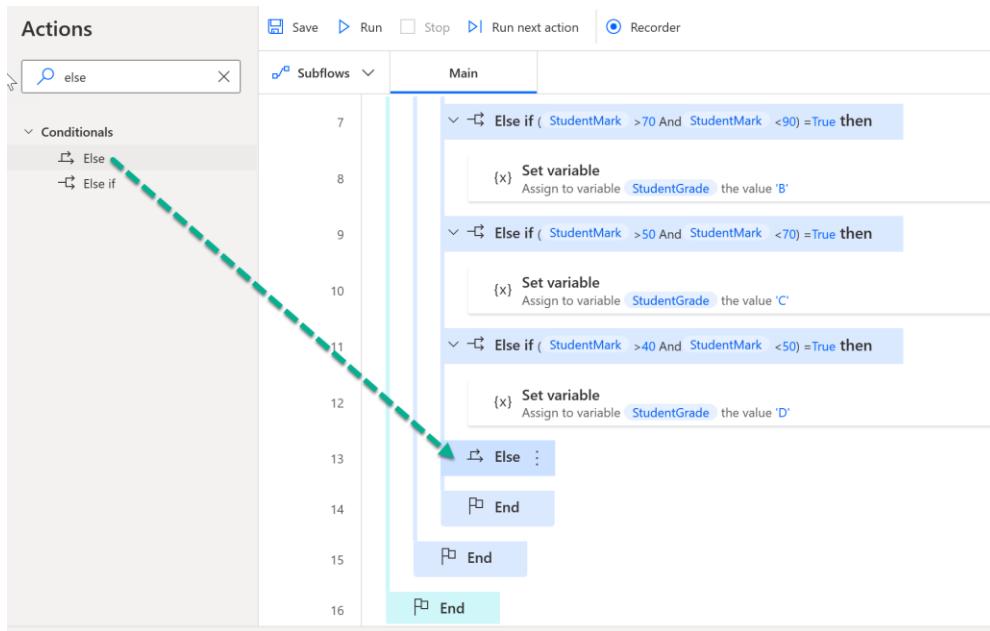
Lets add one last Else If to the nested Ifs and this time we will check the mark to be greater than 40 and less than 50 and evaluate it to True



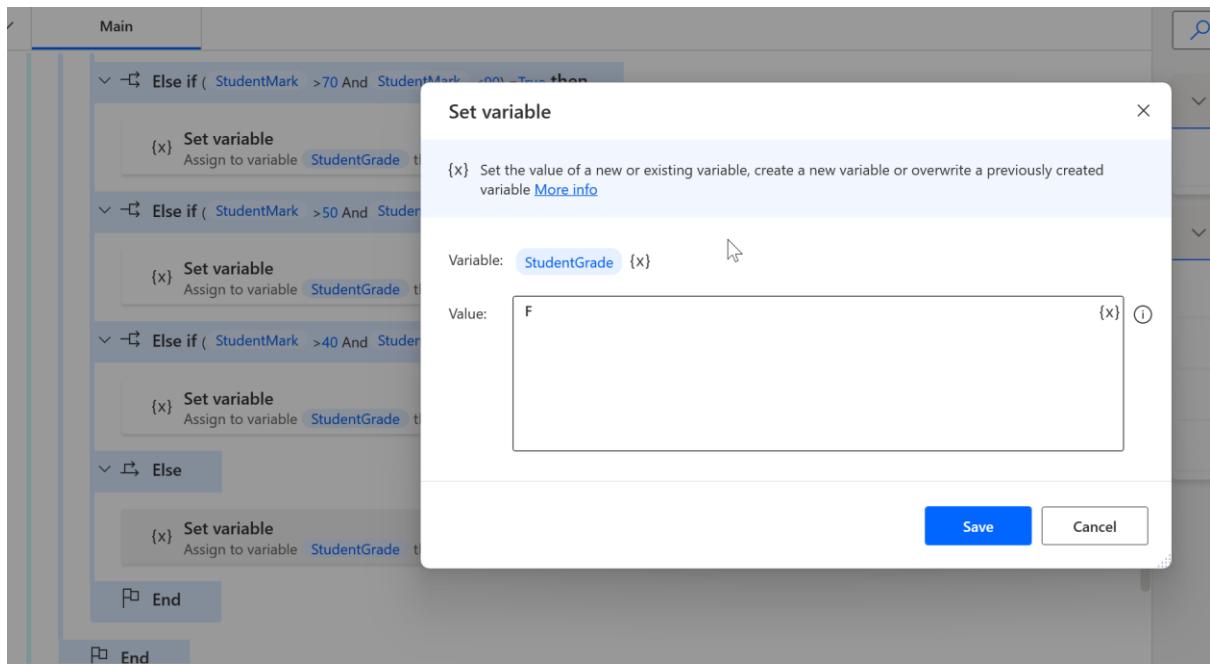
Inside the Else If, we will do the Grade setting to the variable



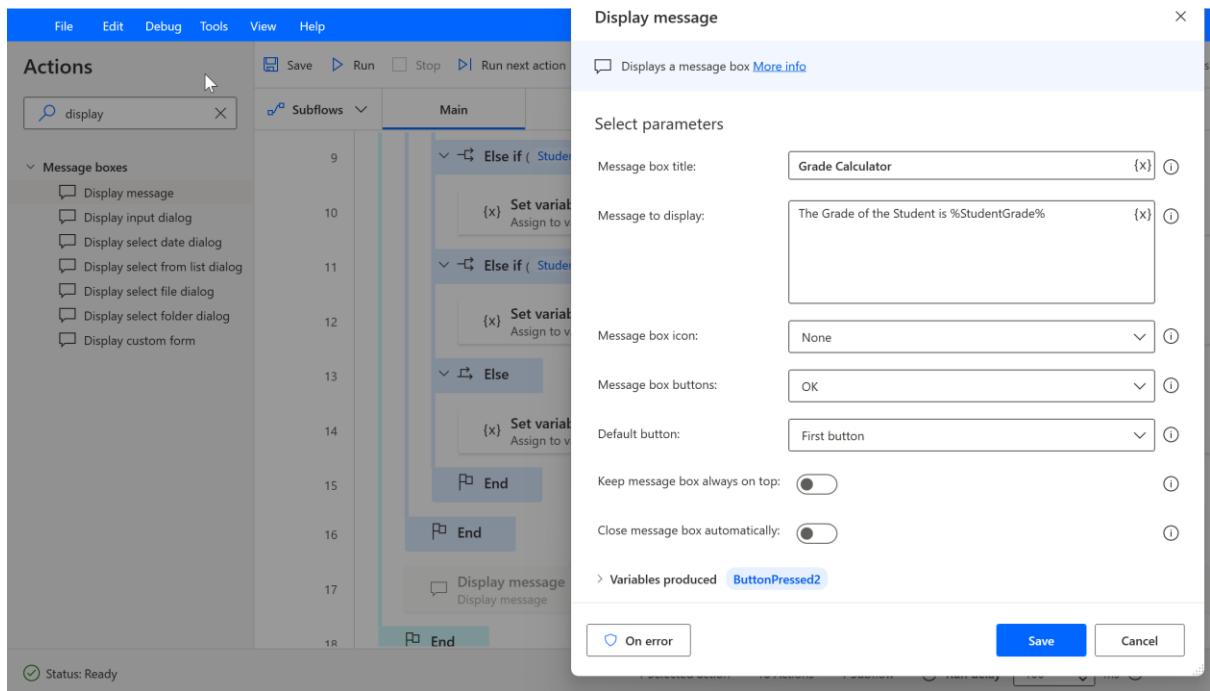
Lets close off the Else if with an else option so that any mark less than 40 can be captured with this condition



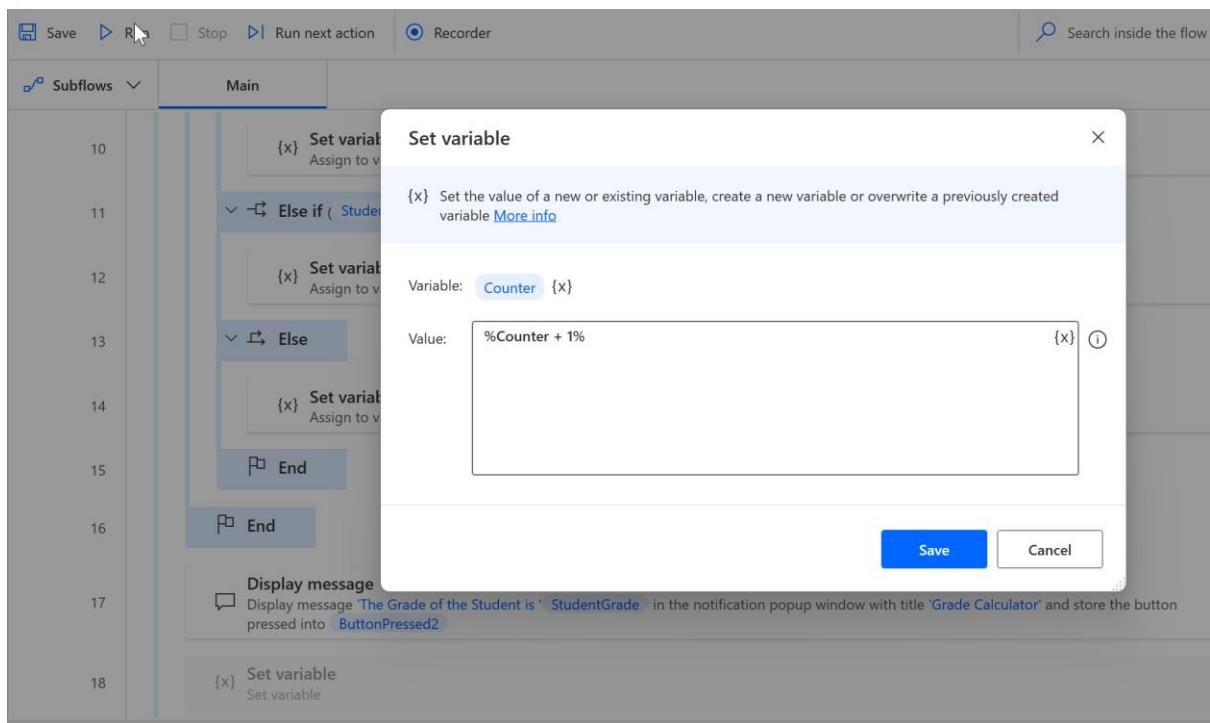
Add the final variable setting for the Failed scenario where we will assign an F to the student



For each student, we need to display the Grade inside the loop for which we will use the Display Message action. In the Message to display field, we will add the StudentGrade variable that holds the analysed grade

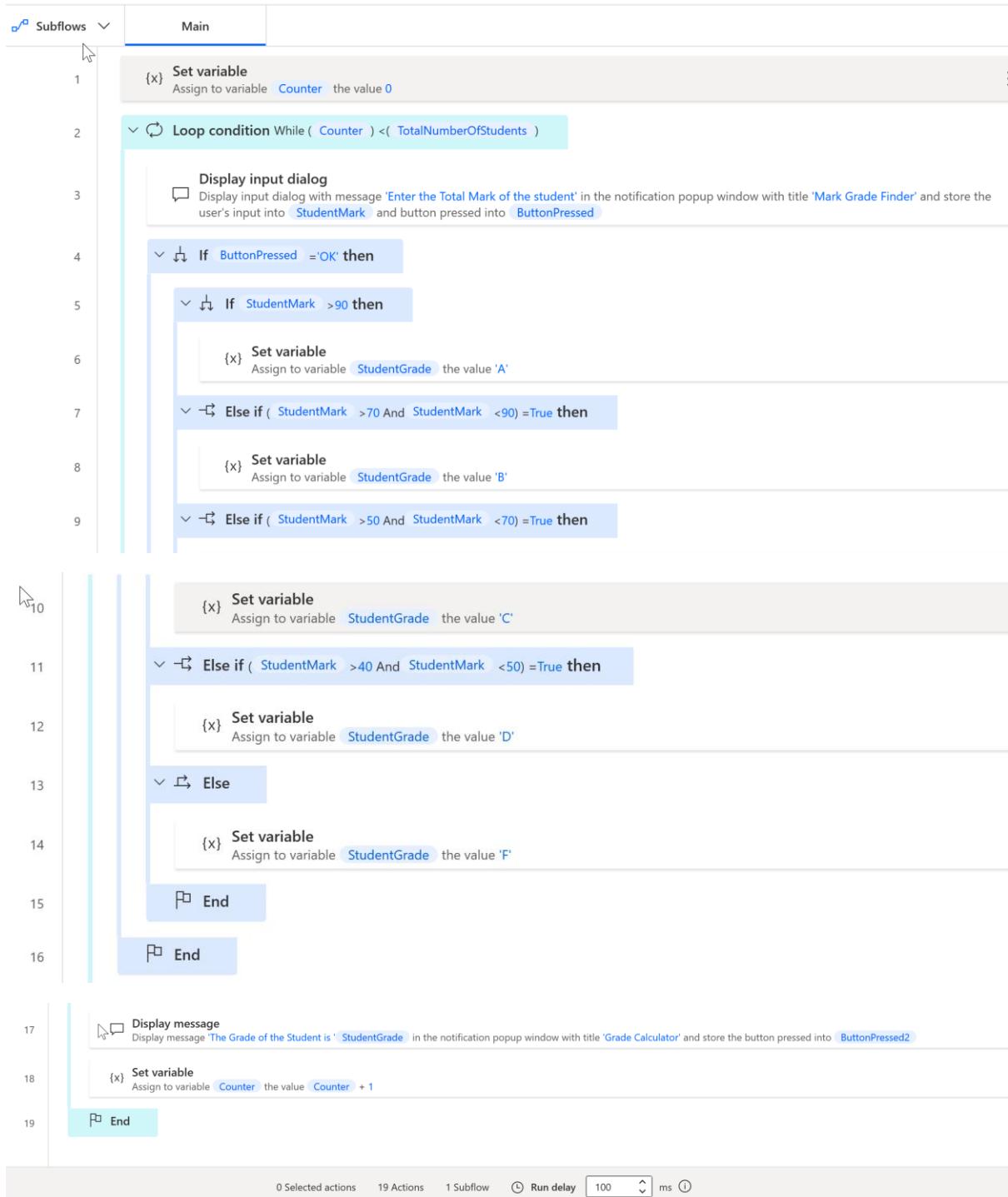


As a last step, we need to increment the loop counter for which we will add the Set variable action

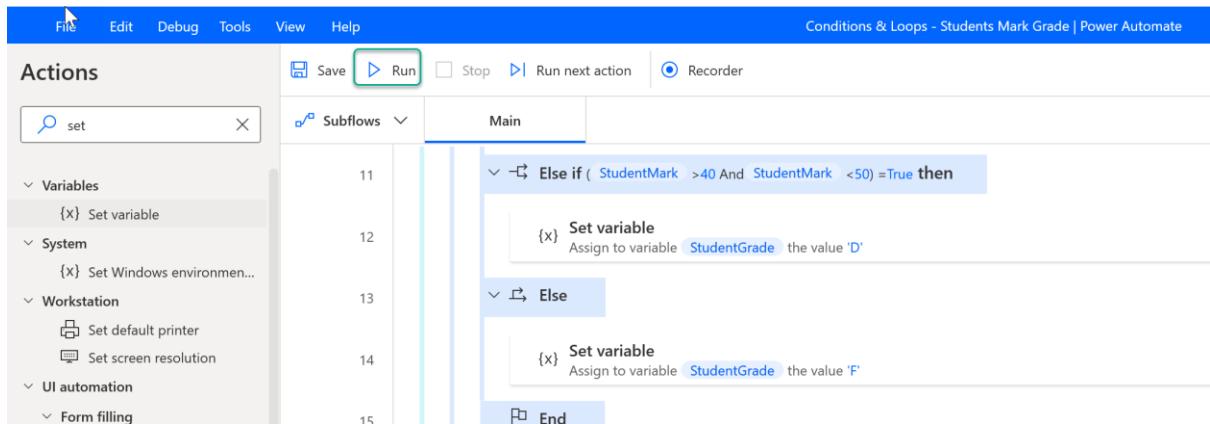


Run the flow

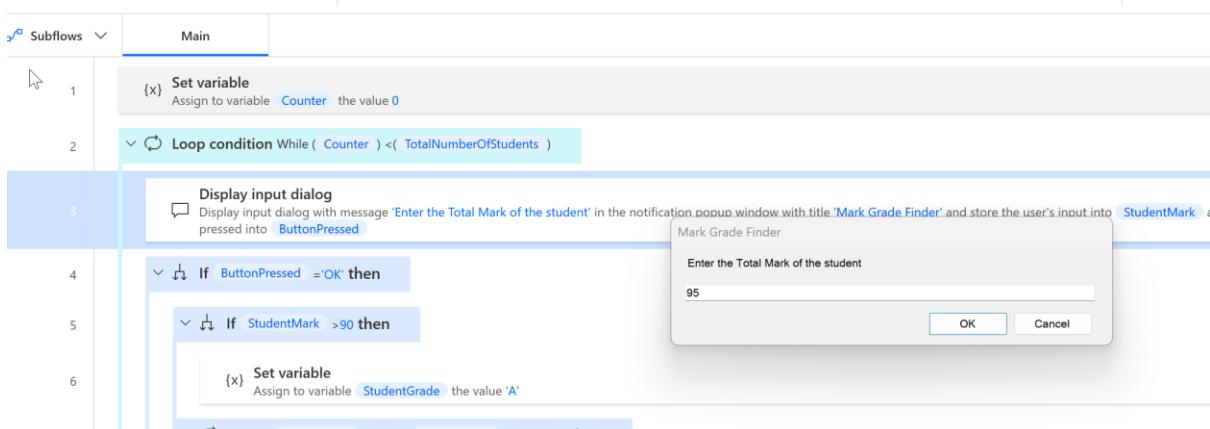
Thus we have completed the creation of the flow which contains 19 steps/actions.



Lets run the flow by clicking on Run



As the flow runs, it asks for the Total marks of the student



Clicking on Ok, will do the conditional analysis and provide us the Grade Output



This loop will continue to run for 5 times as we had defined the Student Count as 5 and once the counter value get incremented after each iteration on the 6th loop iteration check, the loop will exit.

