ADVANCE EXCEL ASSIGNMENT 18

1. **What are comments and what is the importance if commenting in any code?**

   Commenting involves placing Human Readable Descriptions inside of computer programs detailing what the Code is doing. Proper use of commenting can make code maintenance much easier, as well as helping make finding bugs faster. Further, commenting is very important when writing functions that other people will use.

   All programs should be commented in such a manner as to easily describe (in English) the purpose of the code and any algorithms used to accomplish the purpose. A user should be able to utilize a previously written program (or function) without ever having to look at the code, simply by reading the comments. Commenting is the "art" of describing what your program is going to do in "high level" English statements. Commenting is best done before actually writing the code for your program.

2. **What is Call Statement and when do you use this statement?**

   Transfers control to a Function, Sub, or dynamic-link library (DLL) procedure.

   Syntax: [Call] procedureName [(argumentList)]

   where,

   procedureName: Required. Name of the procedure to call.

   argumentList: Optional. List of variables or expressions representing arguments that are passed to the procedure when it is called. Multiple arguments are separated by commas. If you include argumentList, you must enclose it in parentheses.

   You can use the Call keyword when you call a procedure. For most procedure calls, you aren’t required to use this keyword. You typically use the Call keyword when the called expression doesn’t start with an identifier. Use of the Call keyword for other uses isn't recommended. If the procedure returns a value, the Call statement discards it.

3. **How do you compile a code in VBA? What are some of the problem that you might face when you don’t compile a code?**

   Before running your code, the VBA Editor compiles the code. This basically means that VBA examines your code to make sure that all the requirements are there to run it correctly – it will check that all the variables are declared (if you use Option Explicit which you should!), check that all the procedures are declared, check the loops and if statements etc. By compiling the code, VBA helps to minimize any runtime errors occurring.

   Types of VBA Errors:-

a) Syntax errors – A specific line of code is not written correctly

b) Compile errors – Issues that happen when putting together lines of code, though the individual lines of code seem to make sense

c) Runtime errors – When the code is usually correct in principle, but an action taken by the user or the data being used leads to unexpected errors.

4. What are hot keys in VBA? How can you create your own hot keys?

A hotkey is a button that triggers a single action such as opening a folder, executing an application, or stopping a song that's playing. It is a key or a combination of keys on a computer keyboard that, when pressed at one time, performs a task (such as starting an application) more quickly than by using a mouse or other input device. Hot keys are sometimes called shortcut keys.

When assigning a shortcut key remember the letter you press is case sensitive. The default uses the Ctrl key. You can hold down the Shift key when selecting your key.

This does not have to be done when you record the macro and can easily be done afterwards.

You can add either a Shortcut Key or a Description to your macro by selecting (Tools > Macro > Macros) and selecting the "Options" button at the bottom.

If you add a description before recording a macro then this description will appear as a comment on the first few lines of the recorded VBA code.

Remember that any key combinations you assign to macros will take precedence over any built-in shortcut keys.

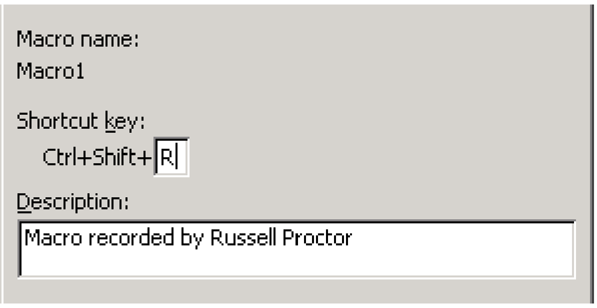
It is important to realize there is a good chance you will be overwriting an existing shortcut key so test first to make sure you will not miss it. Avoid overwriting common shortcut keys such as (Ctrl + "C" or Ctrl + "V").

If you enter "b" in lowercase, you can execute the macro by pressing (Ctrl + "b"). If you enter "B" in uppercase, you can execute the macro by pressing (Ctrl + Shift + "b").

You can add or change a short-cut key at any time. You don't have to set this before recording your macro.

The Descriptive box can be used to add a short description to your macros and is only visible in the (Tools > Macros > Macro) dialog box when the macro is selected.

You could run your macro by using a shortcut key. Select (Tools > Macro > Macros) to display the Macro dialog box. Select the macro that you want to assign a shortcut key to and press the "Options" button.



Macro name:
Macro1

Shortcut key:
Ctrl+Shift+ R

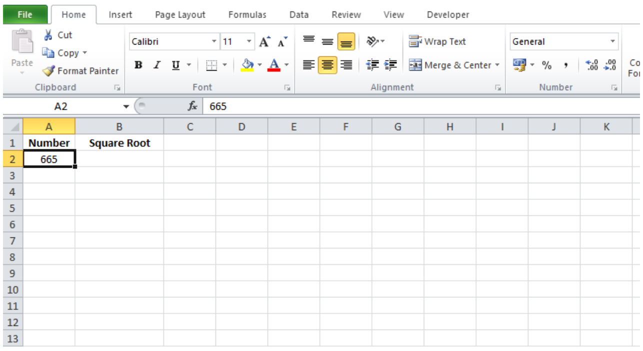
Description:
Macro recorded by Russell Proctor

The shortcut key must be a letter. If you forget to assign a shortcut key you can always do it at a later date by using the Options button on the (Tools > Macro > Macros) dialog box.

5. **Create a macro and shortcut key to find the square root of the following numbers 665, 89, 72, 86, 48, 32, 569, 7521**

Follow these steps to create a macro:

a. Open MS Excel and enter one number to find the square root in the cell for eg:A2 as shown in the figure below. Also, select that cell before recording the macro.

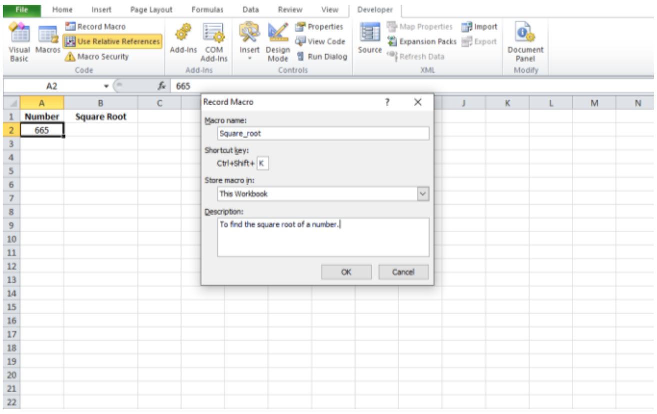


A2 665

	A	B
1	Number	Square Root
2	665	

b. Go to Developer Tab and enable the “Use Relative References” option in the Code group. This will make the macro to run irrespective of the cells we choose at the time of recording the macro.

c. To start recording, click on “Record Macro” which will open a dialogue box in which you need to fill up the details of the macro like name, shortcut key,ect. as shown below.



Record Macro

Macro name: Square_root

Shortcut key: Ctrl+Shift+ K

Store macro in: This Workbook

Description: To find the square root of a number.

d. The macro will start recording from the moment you clicked the OK button in the dialogue box shown above and from now onwards whatever we do in the excel sheet.

e. Initially, select the cell where we need to get the square root of the number and write the below formula in the formula bar and press Enter.

=ROUND(SQRT(A2),2)   where A2 is the cell that contains the input number.

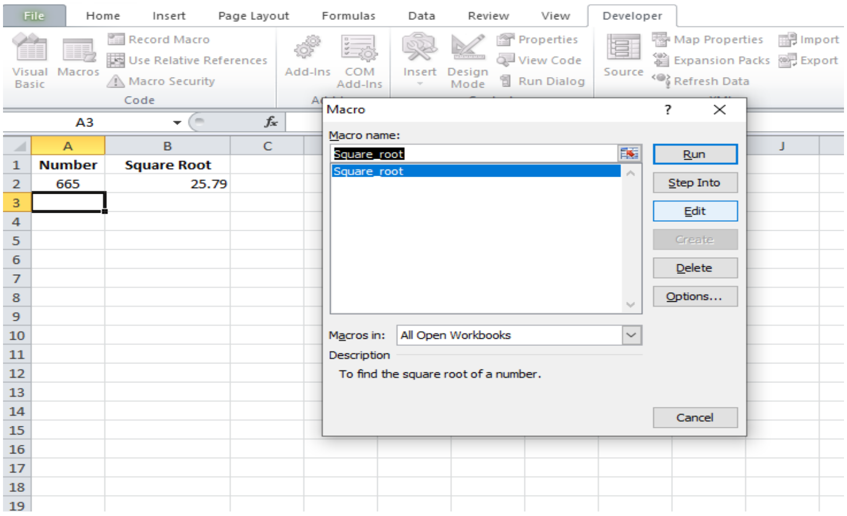
f. Now the result will be shown in that cell and the cursor will be moved to the next cell that is right below in the same column.

g. In order to repeat the process select the next input cell and click “Stop Recording”.

h. Now, the macro is created for only one number.

i. If you need to find the square root of N numbers at once you need to manually run the macro multiple times repeatedly.

j. In order to avoid the repeated execution of macros manually, you can edit the macro code using Do-While loop as shown below by specifying the condition for the termination of the execution:



	A	B
1	Number	Square Root
2	665	25.79
3		

Macro name: Square_root

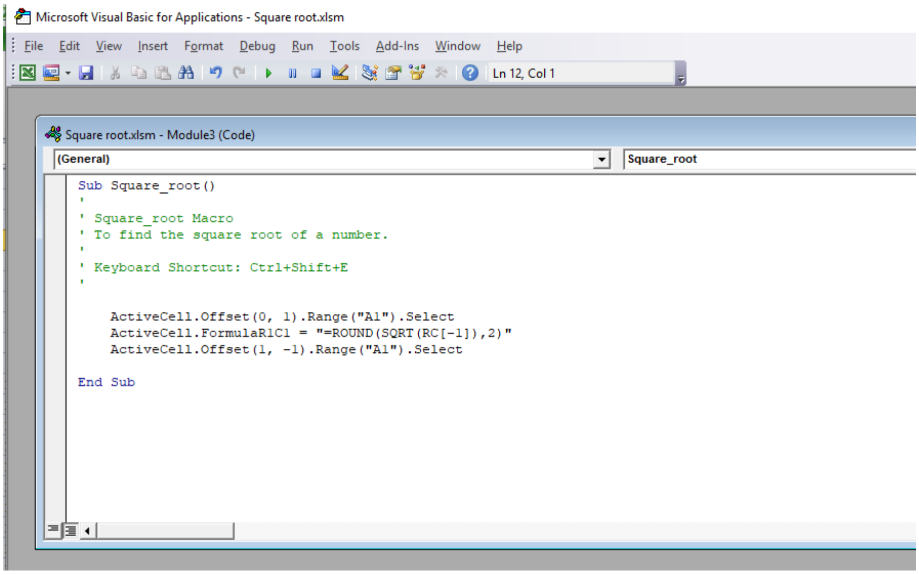
Square_root

Macros in: All Open Workbooks

Description

To find the square root of a number.

Select the macro to Edit.

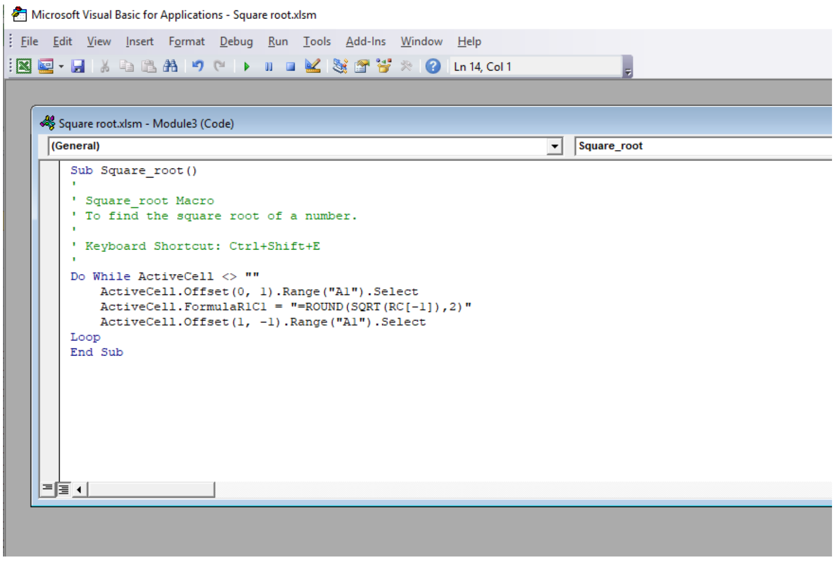

Microsoft Visual Basic for Applications - Square root.xlsm
File
Edit
View
Insert
Format
Debug
Run
Tools
Add-Ins
Window
Help
Ln 12, Col 1
Square root.xlsm - Module3 (Code)
(General)
Square_root

```
Sub Square_root()
'
' Square_root Macro
' To find the square root of a number.
'
' Keyboard Shortcut: Ctrl+Shift+E
'

    ActiveCell.Offset(0, 1).Range("A1").Select
    ActiveCell.FormulaR1C1 = "=ROUND(SQRT(RC[-1]),2)"
    ActiveCell.Offset(1, -1).Range("A1").Select

End Sub
```

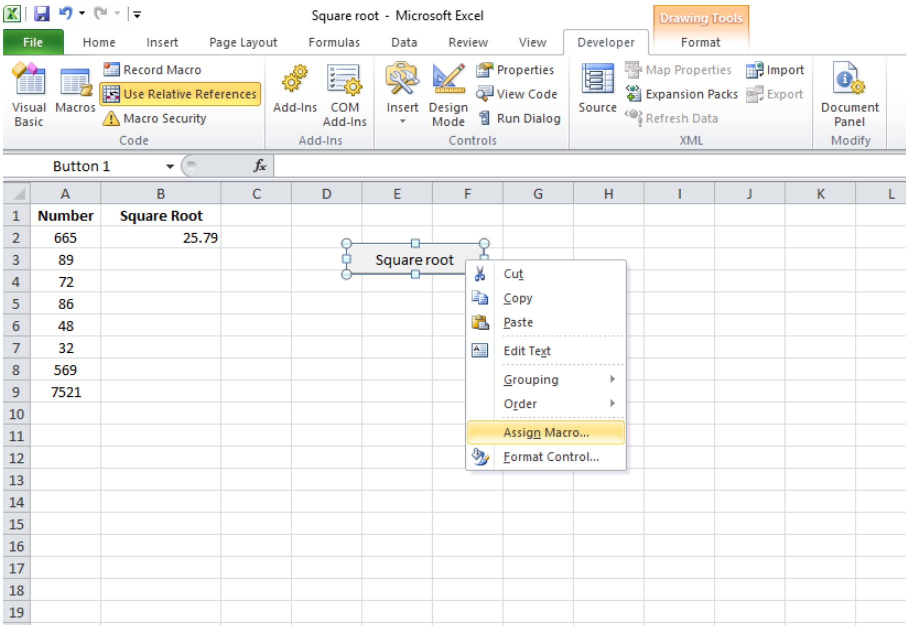
Original code before editing.


Microsoft Visual Basic for Applications - Square root.xlsm
File Edit View Insert Format Debug Run Tools Add-Ins Window Help
Ln 14, Col 1
Square root.xlsm - Module3 (Code)
(General)
Square_root

```
Sub Square_root()
'
' Square_root Macro
' To find the square root of a number.
'
' Keyboard Shortcut: Ctrl+Shift+E
'
Do While ActiveCell <> ""
    ActiveCell.Offset(0, 1).Range("A1").Select
    ActiveCell.FormulaR1C1 = "=ROUND(SQRT(RC[-1]),2)"
    ActiveCell.Offset(1, -1).Range("A1").Select
Loop
End Sub
```

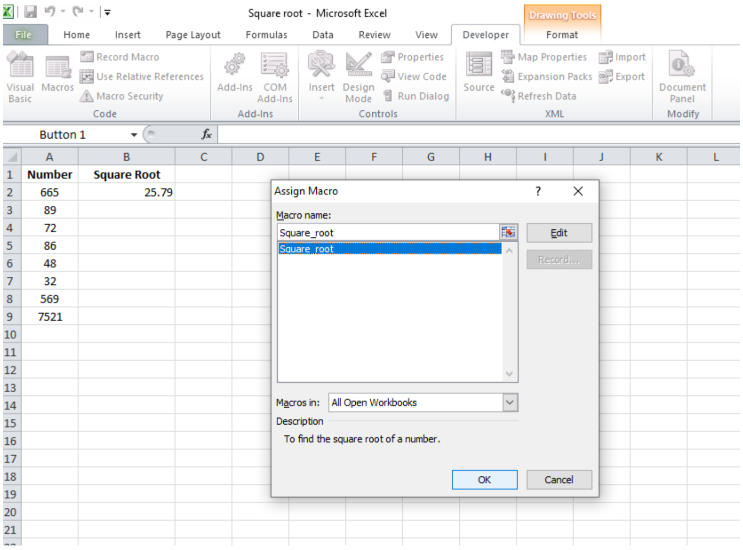
Code after adding while loop to the macro.

k. To make it user friendly, you can add button for the macro in the sheet using Insert Controls in the developer tab and assign the macro by choosing the option "Assign Macro" by right clicking the button. Select the macro you want to assign and click OK.



	A	B
1	Number	Square Root
2	665	25.79
3	89	
4	72	
5	86	
6	48	
7	32	
8	569	
9	7521	

Create the button named "square root".



	A	B
1	Number	Square Root
2	665	25.79
3	89	
4	72	
5	86	
6	48	
7	32	
8	569	
9	7521	

Assign Macro

Macro name: Square_root

Square_root

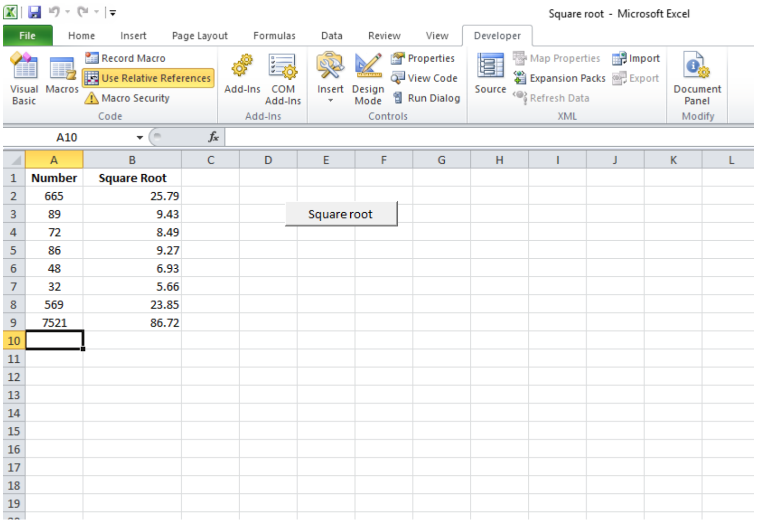
Macros in: All Open Workbooks

Description

To find the square root of a number.

The square root macro is assigned to the button.

l. Now, you can select the cell that contains the input number and click the button to run the macro for a set of numbers provided in the sheet.



	A	B
1	Number	Square Root
2	665	25.79
3	89	9.43
4	72	8.49
5	86	9.27
6	48	6.93
7	32	5.66
8	569	23.85
9	7521	86.72

Final result after running macro on a set of numbers.

6. What are the shortcut keys used to

a. Run the code – F5

b. Step into the code

c. Step out of code

d. Reset the code

a. Run the code: F5 - Runs the current procedure or continues if in Debug Mode (Run > Run Sub).

b. Step into the code: F8 - Step into code line by line. Can be used to start debugging subroutines that require no arguments (Debug > Step Into).

c. Step out of code: Ctrl + Shift + F8 Step out of code (Debug > Step Out).

d. Reset the code : Alt + r + r