Q1. What is a Macro? How is it useful in excel or in your daily work?

Ans: A macro is shorthand for a programming term: macroinstruction. In the most basic terms, a macro is a way to automate simple tasks. They can be used in a variety of ways, but most of us will learn to love them as helpful shortcuts in Microsoft Excel.

A macro is a set of instructions used to execute repetitive tasks. You can record a set of commands and then play them back with one or two keystrokes.

That means that you can save A LOT of time when doing routine and repetitive tasks. Before learning about macros, most of us just resign ourselves to the fact that there will be certain mundane tasks in Excel that we can't escape. But once you discover the power of macros, this can all change. If you've not used macros before you might be completely confused about how they can help you, so let's look at a simple example:

Every week you must update a central budget spreadsheet, pulling in numbers from three different spreadsheets, all with different formatting that you then have to fix.

Every time you do this, you start by opening the main spreadsheet. Then you open spreadsheet one, copy over the data onto the central spreadsheet, and format it so it's in the same style as the rest of the data. And then you do the same with spreadsheets two and three.

It's a boring, monotonous task that could take 15-30 minutes each time.

But if you were to use a macro, you could do all of this automatically with a few clicks. (And watching a macro do the work for you is almost like watching magic happen!)

Q2. What is VBA? Write its full form and briefly explain why VBA is used in excel?

Ans: Visual Basic for Applications (VBA) is part of Microsoft Corporation's (NASDAQ: MSFT) legacy software Visual Basic. VBA is used to write programs for the Windows operating system and runs as an internal programming language in Microsoft Office (MS Office, Office) applications such as Access, Excel, PowerPoint, Publisher, Word, and Visio. VBA allows users to customize beyond what is normally available with MS Office host applications.

All Office suite programs share common programming languages, and each have the capability of integrating VBA code to enhance the program. Due to the repetitive nature of spreadsheets, data analytics, and organizing data, VBA has been a natural fit with Excel more so than other Office suite programs.

At its core, finance is about manipulating huge amounts of data; hence, VBA is endemic to the financial services sector. If you work in finance, VBA is likely running within applications that you use each day, whether you're aware of it or not. Some jobs in the sector require prior knowledge of VBA, and some do not. With VBA, you can:

- **Write macros.** Macros allow financial professionals—whether accountants, commercial bankers, investment bankers, research analysts, salesmen, traders, portfolio managers, clerks, or administrators—to analyze and adjust huge amounts of data quickly.
- **Update data.** You can use VBA in Excel to create and maintain complex trading, pricing, and risk-management models, forecast sales and earnings, and to generate financial ratios.
- **Perform scenario-analysis.** With Visual Basic for Applications, you can create various portfolio-management and investment scenarios. This includes filtering through different situations that may impact outcomes differently.
- **Organize information.** You also may use VBA to produce lists of customers' names or any other content; create invoices, forms, and charts; analyze scientific data, and manage data display for budgets and forecasting.
- **Be unconventional.** VBA can be used to copy and paste values, adjust cell styles for an entire workbook, and strike accelerator keys. You can perform very normal tasks but perform them in an easier, automated manner.
- **Prompt action.** VBA can be used to interact with users. For example, you may need a user's input for their first and last name to put on a form. VBA can be used to prompt a user in a way that makes this input unavoidably mandatory.

Q4. What do you mean when we say VBA Editor?

Ans: The Visual Basic Editor is not exactly the same as Excel. It is actually a separate application, even though you'll usually open it through Excel. In fact, in order for the VBE to be able to run, Excel must be open.

The main function of the VBE is to allow you to write and edit VBA code.

The Visual Basic Editor is sometimes referred to as the Integrated Development Environment (IDE). In this Excel tutorial, I use the first term (Visual Basic Editor or VBE)

but don't be confused if you see the second term being used in other places.

You can open the VBE using either of the following methods:

- Click on “Visual Basic” in the Developer tab of the Ribbon.

- Use the keyboard shortcut “Alt + F11”.

Q5. Briefly describe the interface of a VBA editor? What is properties window? And what is watch window? How do you display these windows?

Ans: Excel’s Visual Basic for Applications (VBA) editor is a very powerful tool.

It lets you write and edit custom scripts that automate actions in Excel**.**

In fact, when you record a macro it is stored in VBA code in the VBA editor.

But writing a macro from the VBA editor directly gives you more flexibility than recording a macro in the traditional manner.

You can create better code and complete more complicated tasks by working directly with Visual Basic for Applications.

Use the Properties window

The Properties window lists the design-time properties for selected objects and their current settings. You can change these properties at design time. When you select multiple controls, the Properties window contains a list of the properties common to all the selected controls.

To navigate the Properties window

1. From the View menu of the Visual Basic Editor, choose Properties window (F4).
2. Select the object whose properties you want to display. You can either use the mouse to select the object or use the Project Explorer to choose from a list.
3. Click the Alphabetic tab to display properties in alphabetic order, or choose the Categorized tab to display object properties by category.

To change a property's value

1. Select the property in the left column.
2. Change the property's value in the right column.

Watch Window

Watch Window in Excel is used to watch for the changes in the formulas we use while working with a large amount of data and formulas. They are available from the “Formulas” tab in the “Formula Auditing” section. When we click on the “Watch Window,” a wizard box appears. It allows us to select the cell for which the values need to be monitored or watched.

To create a watch window in a workbook. Go to Formulas in the MS toolbar, click the “watch window” option.

It will create a blank window. The screenshot for the same is as follows.

Key Point**:** Hotkey to add a watch window is **(**Alt key+ M+ W**)**

As you can see above, the Watch window is added. But there are no cells added to it. To add any cell to the watch window, click on “Add watch” in the window. A new dialog box will appear. In this box, you have to input the location of the cell on whose value you have to keep an eye on.

The cell you have entered in it will be highlighted. At first, you can note the input in a dialog box; it has value pointing towards the worksheet name and cell address.

Q6. What is an immediate Window and what is it used for?

Ans: The Immediate window displays information resulting from debugging statements in your code or from commands typed directly into the window.

To display the Immediate window

- From the View menu, choose Immediate window (CTRL+G).

To execute code in the Immediate window**:**

1. Type a line of code in the Immediate window.
2. Press ENTER to execute the statement**.**

Use the Immediate window to:

- Test problematic or newly written code.

- Query or change the value of a variable while running an application. While execution is halted, assign the variable a new value as you would in code.
- Query or change a property value while running an application.
- Call procedures as you would in code.
- View debugging output while the program is running.