

# How do I find the first 3 positive numbers in Excel when I have a column of numbers?

Authored by  
**stats writer**

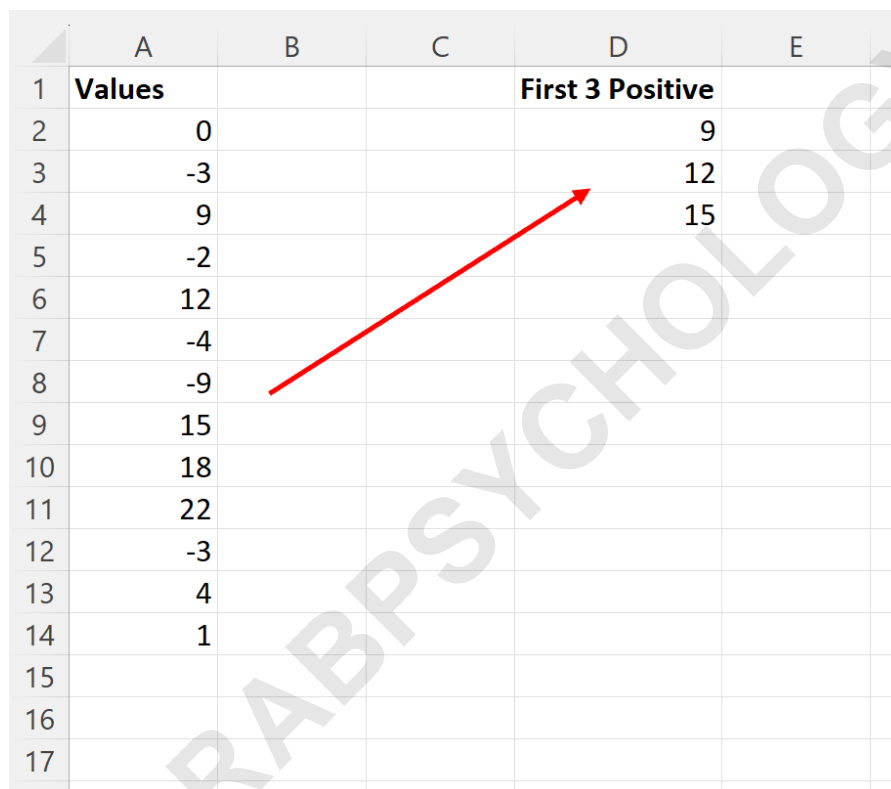
November 18, 2025

## RECOMMENDED CITATION

stats writer (2025). *How do I find the first 3 positive numbers in Excel when I have a column of numbers?*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=95391>

In the realm of data analysis using Excel, practitioners frequently encounter scenarios requiring sophisticated filtering and extraction techniques. A common requirement is the need to identify the initial set of positive numbers appearing sequentially within a larger column of mixed data, which includes negative values and zeros. This necessity arises often in financial modeling, scientific data logging, or performance tracking where chronological order matters greatly.

For instance, imagine you are monitoring daily stock gains or temperature fluctuations. You might have a lengthy column containing various numerical results, and the specific task is to isolate only the first three instances where the recorded value exceeds zero, maintaining their original order of appearance. Traditional filtering or simple sorting methods are insufficient here, as they destroy the sequence of data arrival.



The image shows an Excel spreadsheet with two columns. Column A is labeled 'Values' and contains the following numbers from row 2 to 14: 0, -3, 9, -2, 12, -4, -9, 15, 18, 22, -3, 4, 1. Column D is labeled 'First 3 Positive' and contains the numbers 9, 12, and 15. A red arrow points from the value 15 in row 9 of column A to the value 15 in row 4 of column D. A large watermark 'ARABPSYCHOLOGY.COM' is overlaid diagonally across the spreadsheet.

	A	B	C	D	E
1	<b>Values</b>			<b>First 3 Positive</b>	
2	0			9	
3	-3			12	
4	9			15	
5	-2				
6	12				
7	-4				
8	-9				
9	15				
10	18				
11	22				
12	-3				
13	4				
14	1				
15					
16					
17					

To accurately solve this complex sequential extraction problem within Excel, we must employ a powerful combination of functions, typically relying on an advanced concept known as an array formula. The following detailed example provides a robust methodology for isolating the first three (or any number N) positive numbers from a given column.

## Solving the Challenge: Locating the First N Positive Values in Excel

## Understanding the Goal: Identifying Sequential Positive Values

When working with datasets in a spreadsheet application like Excel, data integrity and ordering are paramount. Simply filtering a list to show values greater than zero fails if you need to know which values were the absolute first few entries that met that criterion. We need a formulaic approach that dynamically checks each cell's position and value simultaneously, returning only the original values corresponding to the earliest positive occurrences.

The solution involves generating an array of relative row positions for all cells that meet our condition (being greater than zero), and then using a lookup function to retrieve the actual value from the original dataset based on those generated positions. This technique ensures that we extract the data points exactly as they appeared in the initial sequence, fulfilling the requirement of finding the "**first**" items.

We will be utilizing three critical functions in harmony: the INDEX function for retrieval, the SMALL function for ranking the positions, and the IF function to conditionally create the necessary array of row numbers.

## Setting Up the Data Environment

To illustrate this method, let us consider a typical list of values entered into column A of an Excel worksheet. This dataset contains a variety of numbers, including both positive numbers, negative numbers, and zeros. Our objective is to populate a new column (Column D) with the first three positive results found in column A.

	A	B	C	D	E	F
1	<b>Values</b>					
2	0					
3	-3					
4	9					
5	-2					
6	12					
7	-4					
8	-9					
9	15					
10	18					
11	22					
12	-3					
13	4					
14	1					
15						
16						
17						
18						

Careful observation of the source data (A2:A14) reveals a mixture of data points. For instance, the data starts with -5, followed by 9, then 0, and so on. The challenge is identifying 9 as the first positive number, then proceeding to the next one in sequence, which requires a precise positional reference.

Before applying the core formula, we need a mechanism to tell the formula which position we are currently looking for (i.e., the 1st positive number, the 2nd positive number, the 3rd positive number, etc.). We achieve this by listing out the ordinal positions (1, 2, 3...) in an adjacent helper column, which we designate as Column C.

	A	B	C	D	E	F
1	<b>Values</b>		1			
2	0		2			
3	-3		3			
4	9					
5	-2					
6	12					
7	-4					
8	-9					
9	15					
10	18					
11	22					
12	-3					
13	4					
14	1					
15						
16						
17						

In this setup, cell C1 contains '1', C2 contains '2', and C3 contains '3'. These numbers will serve as the input for the rank argument within the SMALL function, instructing it to pull the 1st smallest relative row number, 2nd smallest, and 3rd smallest, respectively.

### The Core Formula Breakdown: INDEX, SMALL, and IF

The solution relies on a complex array formula that must be entered into the result cell (D1) and then copied down. This formula performs three key operations simultaneously: testing for positivity, calculating relative position, and retrieving the final value.

The complete formula to be entered into cell **D1** is:

```
=INDEX($A$2:$A$14,SMALL(IF($A$2:$A$14>0,ROW($A$2:$A$14)-ROW($A$2)+1),C1))
```

This formula is powerful because it operates on the entire range \$A\$2:\$A\$14 simultaneously, generating an array of row offsets for every positive number found. Because this is an array calculation, if you are using an older version of Excel (pre-Microsoft 365 or 2021), you must confirm the entry by pressing **CTRL + SHIFT + ENTER**, which wraps the formula in curly braces {}. Newer versions handle these dynamic array computations automatically.

## Analyzing the Components of the Array Formula

To ensure clarity and understanding, let us dissect the inner components of this sophisticated formula, starting from the innermost calculation and working our way out:

**Condition Check ( $\$A\$2:\$A\$14>0$ ):** This section checks every cell in the defined range (A2 through A14) to see if it is greater than zero. It returns an array formula of TRUE or FALSE values.

**Position Calculation ( $\text{ROW}(\$A\$2:\$A\$14)-\text{ROW}(\$A\$2)+1$ ):** The ROW function returns the absolute row number for each cell in the range (e.g., 2, 3, 4...). By subtracting the row number of the starting cell (A2, which is 2) and adding 1, we convert the absolute row numbers into relative positions within the range (e.g., 1, 2, 3... 13). This is crucial for the INDEX function.

**Conditional Array Generation ( $\text{IF}(\$A\$2:\$A\$14>0, )$ ):** The IF function uses the TRUE/FALSE results from the condition check. If a value is positive (TRUE), the IF function returns its relative position (1, 2, 3...); if it is not positive (FALSE), it returns FALSE. This creates a compact array like {FALSE; 2; FALSE; 4; ...}.

**Ranking the Positions ( $\text{SMALL}(\text{IF}(\dots), C1)$ ):** The SMALL function takes the conditional array and extracts the smallest numerical value. Since we reference cell **C1** (which is 1), it finds the 1st smallest relative position that corresponds to a positive number. When copied down to D2 (referencing C2, which is 2), it finds the 2nd smallest position, and so on.

**Final Retrieval ( $\text{INDEX}(\$A\$2:\$A\$14, \text{SMALL}(\dots))$ ):** Finally, the INDEX function uses the calculated relative position retrieved by SMALL to look up and return the actual value from the source range \$A\$2:\$A\$14.

## Applying the Formula to Find the First Three Results

Once the formula is correctly entered into cell **D1** (remembering to use **CTRL + SHIFT + ENTER** if necessary), it immediately calculates the position of the first positive value in the column and returns that value. In our specific dataset, the first positive value (9) is located at the second relative position within the range A2:A14 (A3).

To find the second and third positive values, we simply click and drag the formula down from cell D1 to cells **D2** and **D3**. Because the reference to the rank argument (C1) is relative, it automatically adjusts to C2 (for the 2nd smallest position) and C3 (for the 3rd smallest position) in the subsequent cells.

	A	B	C	D	E	F	G
1	Values		1	9			
2	0		2	12			
3	-3		3	15			
4	9						
5	-2						
6	12						
7	-4						
8	-9						
9	15						
10	18						
11	22						
12	-3						
13	4						
14	1						
15							
16							
17							

The results in Column D now clearly display the first three positive numbers extracted from column A in their order of appearance: **9**, **12**, and **15**. Any negative numbers or zeros encountered before these values were correctly ignored by the IF condition within the array formula structure, preserving the integrity of the sequence.

### Extending the Solution: Finding the First 'N' Positive Numbers

A key advantage of this array formula technique is its scalability. It is not limited strictly to finding only three values; you can use this exact same setup to find the first  $n$  positive values in a column, where  $n$  could be any number up to the total count of positive values available.

If we wished to list out the first five positive values from column A, we would simply extend the list in Column C down to 5 (C5) and then drag our formula from D3 down to D5. The logic remains sound, as the SMALL function would then be instructed to find the 4th and 5th smallest relative positions.

	A	B	C	D	E	F
1	<b>Values</b>		1	9		
2	0		2	12		
3	-3		3	15		
4	9		4	18		
5	-2		5	22		
6	12					
7	-4					
8	-9					
9	15					
10	18					
11	22					
12	-3					
13	4					
14	1					
15						
16						
17						
18						

As demonstrated above, Column D now correctly displays 9, 12, 15, 22, and 45. If the formula were dragged down past the point where all positive numbers have been found, it would typically result in a **#NUM!** error, indicating that the SMALL function could not find a corresponding position.

### Conclusion: Versatility and Power of Array Formulas

Mastering complex sequential filtering tasks in Excel, such as isolating the first  $n$  positive values, significantly enhances data processing capabilities. By leveraging the combined strengths of the INDEX function and the SMALL function within an array formula environment, we achieve a dynamic and highly efficient filtering solution that respects the chronological order of the data.

This method provides a robust alternative to manual sorting and standard filtering, especially when dealing with large datasets where maintaining the original sequence is vital for meaningful analysis. We encourage users to adapt this formula structure--modifying the condition (e.g., finding the first  $n$  negative values, or values greater than a specific threshold) and range references--to suit the diverse requirements of their own datasets.

This powerful technique is a testament to the flexibility and depth available within Excel when advanced functions are properly utilized.