

# 2469. Convert the Temperature

Easy Math

[Leetcode Link](#)

## Problem Description

In this problem, you are given a non-negative floating point number `celsius`, which represents a temperature in Celsius. This value is rounded to two decimal places for precision. Your task is to convert this Celsius temperature to two other temperature scales: Kelvin and Fahrenheit.

To return your answers, you will need to create an array `ans = [kelvin, fahrenheit]`, where `kelvin` is the temperature in Kelvin and `fahrenheit` is the temperature in Fahrenheit.

It is important to remember two key conversion formulas:

1. To convert Celsius to Kelvin, you add 273.15 to the Celsius temperature.
2. To convert Celsius to Fahrenheit, you multiply the Celsius temperature by 1.80 and then add 32.00 to the result.

The problem specifies that your answers will be accepted as long as they are within  $10^{-5}$  of the correct answer, allowing for a very small amount of numerical error which is common when dealing with floating-point arithmetic.

## Intuition

The intuition behind the solution is straightforward since the problem simply involves applying the given conversion formulas. There are no complex algorithms or data structures involved, just basic arithmetic.

To solve the problem, follow these steps:

1. Add 273.15 to the `celsius` value to get the Kelvin temperature.
2. Multiply the `celsius` value by 1.8 and then add 32 to get the Fahrenheit temperature.

These steps are based directly on the conversion formulas given in the problem description. Once you have both converted values, you create an array containing the Kelvin and Fahrenheit temperatures in that order and return it. This is made simple in Python with list notation.

Because the problem involves real numbers and potentially floating-point precision issues, it's important to ensure that calculations are performed with enough precision to meet the requirement of being within  $10^{-5}$  of the actual value. However, Python's built-in floating-point arithmetic should be precise enough for this task without needing any specialized numerical techniques.

## Solution Approach

The solution approach is remarkably direct and does not involve any complex algorithms or use of advanced data structures. It adheres to the simplicity of the problem statement, which requires basic arithmetic operations based on the formulas given for temperature conversion. Here's how the provided solution in Python implements the approach step by step:

1. Define a class `Solution` and a method `convertTemperature` within it. The method accepts one parameter `celsius`, which is a floating-point number representing the temperature in Celsius.
2. Inside the method, perform the arithmetic conversions as per the formulas given in the problem description:
  - Convert Celsius to Kelvin by adding 273.15 to the `celsius` value. This relies on the formula `Kelvin = Celsius + 273.15`.
  - Convert Celsius to Fahrenheit by multiplying the `celsius` value by 1.8 and then adding 32. This follows the formula `Fahrenheit = Celsius * 1.80 + 32.00`.
3. Return the results of the conversions in an array `[kelvin, fahrenheit]`. In Python, this is accomplished by returning the values in a list.

The solution does not use any complex data structures. It only requires a simple list to store the two resulting temperature values. No patterns or algorithms are needed; the problem is solved by directly applying the given mathematical conversion formulas.

Here's what the implementation looks like in code:

```
1 class Solution:
2     def convertTemperature(self, celsius: float) -> List[float]:
3         kelvin = celsius + 273.15 # Convert Celsius to Kelvin
4         fahrenheit = celsius * 1.8 + 32 # Convert Celsius to Fahrenheit
5         return [kelvin, fahrenheit] # Return the results in an array
```

Note that in the code snippet, the conversion operations have been spelled out in separate statements for clarity, although the original solution provided performs the calculations inline within the return statement. Both approaches are functionally equivalent and produce the same result.

## Example Walkthrough

Let's illustrate the provided solution approach with a small example. Suppose we have a temperature in Celsius that is 25.00 degrees. We want to convert this to Kelvin and Fahrenheit using the formulas from the problem description.

1. The formula to convert Celsius to Kelvin is: `Kelvin = Celsius + 273.15`. So we start with our Celsius value 25.00 and add 273.15 to it.

```
1 Kelvin = 25.00 + 273.15
2 Kelvin = 298.15
```

2. The formula to convert Celsius to Fahrenheit is: `Fahrenheit = Celsius * 1.80 + 32.00`. We take the same original Celsius value and apply the formula.

```
1 Fahrenheit = 25.00 * 1.80 + 32
2 Fahrenheit = 45.00 + 32
3 Fahrenheit = 77.00
```

So with our input `celsius` being 25.00 degrees, after the conversion, we have 298.15 Kelvin and 77.00 Fahrenheit. We then create an array with these two values in the order described: `[kelvin, fahrenheit]`.

Following the solution approach, our result array would look like this:

```
1 ans = [298.15, 77.00]
```

This array is what we would expect the `convertTemperature` method of the `Solution` class to return when it is passed 25.00 as the `celsius` parameter.

The complete code would look like this:

```
1 class Solution:
2     def convertTemperature(self, celsius: float) -> List[float]:
3         kelvin = celsius + 273.15 # Convert Celsius to Kelvin
4         fahrenheit = celsius * 1.8 + 32 # Convert Celsius to Fahrenheit
5         return [kelvin, fahrenheit] # Return the results in an array
```

And calling the method would look like this:

```
1 sol = Solution()
2 print(sol.convertTemperature(25.00)) # Output will be [298.15, 77.00]
```

The output of the code accurately reflects the calculated Kelvin and Fahrenheit temperatures based on the Celsius input, and this walkthrough demonstrates how the solution approach is followed to arrive at the result.

## Python Solution

```
1 # Import the List type from the typing module for type hinting.
2 from typing import List
3
4 class Solution:
5     # Method to convert temperature from Celsius to Kelvin and Fahrenheit.
6     def convertTemperature(self, celsius: float) -> List[float]:
7         # Convert Celsius to Kelvin by adding 273.15.
8         kelvin = celsius + 273.15
9         # Convert Celsius to Fahrenheit by multiplying by 1.8 (or 9/5) and adding 32.
10        fahrenheit = celsius * 1.8 + 32
11        # Return the temperatures in Kelvin and Fahrenheit as a list of floats.
12        return [kelvin, fahrenheit]
13
```

## Java Solution

```
1 class Solution {
2     public double[] convertTemperature(double celsius) {
3         // Create an array to hold the converted temperatures
4         double[] convertedTemperatures = new double[2];
5
6         // Convert Celsius to Kelvin and store the result in the first element of the array
7         // The formula to convert Celsius to Kelvin is celsius + 273.15
8         convertedTemperatures[0] = celsius + 273.15;
9
10        // Convert Celsius to Fahrenheit and store the result in the second element of the array
11        // The formula to convert Celsius to Fahrenheit is celsius * 1.8 + 32
12        convertedTemperatures[1] = celsius * 1.8 + 32;
13
14        // Return the array containing both converted temperatures
15        return convertedTemperatures;
16    }
17 }
18
```

## C++ Solution

```
1 #include <vector> // Include the header for using the vector container
2
3 // Define a Solution class
4 class Solution {
5 public:
6     // Function to convert Celsius to both Kelvin and Fahrenheit
7     // Takes one double as an argument representing the Celsius temperature
8     // Returns a vector of doubles containing both the Kelvin and Fahrenheit conversions
9     vector<double> convertTemperature(double celsius) {
10        // Declare a vector to hold the converted temperatures
11        vector<double> convertedTemperatures;
12
13        // Convert Celsius to Kelvin and add to the vector
14        convertedTemperatures.push_back(celsius + 273.15); // Kelvin conversion
15
16        // Convert Celsius to Fahrenheit and add to the vector
17        convertedTemperatures.push_back(celsius * 1.8 + 32); // Fahrenheit conversion
18
19        // Return the vector containing the Kelvin and Fahrenheit temperatures
20        return convertedTemperatures;
21    }
22 };
23
```

## Typescript Solution

```
1 // Converts a temperature from Celsius to Kelvin and Fahrenheit.
2 // @param celsius - The temperature in Celsius to convert.
3 // @return An array containing the temperature in Kelvin and Fahrenheit.
4 function convertTemperature(celsius: number): number[] {
5     // Convert Celsius to Kelvin
6     const kelvin: number = celsius + 273.15;
7
8     // Convert Celsius to Fahrenheit
9     const fahrenheit: number = celsius * 1.8 + 32;
10
11    // Return both converted temperatures as an array
12    return [kelvin, fahrenheit];
13 }
14
```

## Time and Space Complexity

### Time Complexity

The time complexity of the `convertTemperature` function is  $O(1)$ , which is constant time complexity. This is because the function does only a fixed number of mathematical operations (two operations in this case) that do not depend on the size of the input.

### Space Complexity

The space complexity of the `convertTemperature` function is also  $O(1)$ . The function creates a list with two elements every time it is called, and the size of this list does not depend on the input size but is fixed.