

## Practice Question

**Q1. Write a Python program to create an attendance sheet. The program should take each student's name and their attendance status (Present/Absent) as input. It should then display the total number of students who are present and absent, save the attendance record, and allow users to check the attendance status of a specific student.**

**Q2. Write a temperature-conversion program that gives the user the option of converting Fahrenheit to Celsius or Celsius to Fahrenheit. Then carry out the conversion. Use floating-point numbers. Interaction with the program might look like this:**

1. to convert Fahrenheit to Celsius,
2. to convert Celsius to Fahrenheit: 1

Enter temperature in Fahrenheit: 70  
In Celsius that's 21.111111

**Q3. Assume that you want to generate a table of multiples of any given number. Write a program that allows the user to enter the number and then generates the table, formatting it into 10 columns and 20 lines. Interaction with the program should look like this (only the first three lines are shown):**

Enter a number: 7  
7 14 21 28 35 42 49 56 63 70  
77 84 91 98 105 112 119 126 133 140  
147 154 161 168 175 182 189 196 203 210

**Q4. Write a program that calculates how much money you'll end up with if you invest an amount of money at a fixed interest rate, compounded yearly. Have the user furnish the initial amount, the number of years, and the yearly interest rate in percent. Some interaction with the program might look like this:**

Enter initial amount: 3000  
Enter number of years: 10  
Enter interest rate (percent per year): 5.5  
At the end of 10 years, you will have 5124.43 dollars.

At the end of the first year you have  $3000 + (3000 * 0.055)$ , which is 3165. At the end of the second year you have  $3165 + (3165 * 0.055)$ , which is 3339.08. Do this as many times as there are years. A for loop makes the calculation easy

## Practice Question

- Q5. Write a Python program that takes an integer input from the user and returns its reversed form. The program should ensure that the input number is greater than 0 and less than 1000. If the number does not meet this condition, display an appropriate error message. For example, if the user enters 123, the program should output 321.**
- Q6. Write a Python program that finds the second largest number in a list without sorting it. The program should iterate through the list and determine the second largest number using a single pass. Avoid using built-in sorting functions.**