

# PHP PROGRAMMING

## PROJECT REPORT

Practical Implementation of PHP Programs

<b>Subject:</b>	PHP Programming
<b>Total Tasks:</b>	2
<b>Date:</b>	November 2025

# TASK 1

## Task Name

Find the Largest of Three Numbers Using Nested If

## AIM

To write a PHP program that accepts three numbers as input and determines the largest among them using nested if-else statements.

## Problem Statement

Given three numbers, we need to compare them and find which one is the largest. The comparison should be done using nested if-else conditional statements to demonstrate the control flow in PHP programming.

## Constraints

- The program accepts three numeric values (integer or float)
- Comparison must be performed using nested if statements only
- The program should handle equal numbers correctly

## Procedure

1. Declare and initialize three variables with numeric values
2. Display the input numbers
3. Use the first if statement to compare the first two numbers
4. Use nested if statements to compare with the third number
5. Store the largest number in a variable
6. Display the largest number as output

## Program

```
<?php
// Task 1: Find the largest of three numbers using nested if

echo "=== Finding Largest of Three Numbers ===\n\n";

// Input three numbers
$num1 = 45;
$num2 = 78;
$num3 = 62;
```

```
echo "Number 1: $num1\n";
echo "Number 2: $num2\n";
echo "Number 3: $num3\n\n";

// Find largest using nested if
if ($num1 >= $num2) {
    if ($num1 >= $num3) {
        $largest = $num1;
    } else {
        $largest = $num3;
    }
} else {
    if ($num2 >= $num3) {
        $largest = $num2;
    } else {
        $largest = $num3;
    }
}

echo "The largest number is: $largest\n";
?>
```

## Output

```
=== Finding Largest of Three Numbers ===

Number 1: 45
Number 2: 78
Number 3: 62

The largest number is: 78
```

## Conclusion

The PHP program successfully determines the largest of three numbers using nested if-else statements. The program demonstrates proper use of conditional logic and control flow. In this execution, among the three numbers 45, 78, and 62, the program correctly identified 78 as the largest number. This approach can be extended to handle user input and validate data types for robust applications.

# TASK 2

## Task Name

Reverse a String Using strrev() Function

## AIM

To write a PHP program that reverses a given string using the built-in strrev() function.

## Problem Statement

Given a string input, we need to reverse the order of characters in the string. PHP provides a built-in function strrev() that performs this operation efficiently. The task is to demonstrate the usage of this function with a sample string.

## Constraints

- The input must be a valid string
- The strrev() function must be used for reversal
- Both original and reversed strings should be displayed

## Procedure

1. Declare and initialize a string variable with a text value
2. Display the original string
3. Use the strrev() function to reverse the string
4. Store the reversed string in a new variable
5. Display the reversed string as output

## Program

```
<?php
// Task 2: Reverse a string using strrev()

echo "=== String Reversal using strrev() ===\n\n";

// Input string
$originalString = "Hello World PHP Programming";

echo "Original String: $originalString\n";

// Reverse the string using strrev()
$reversedString = strrev($originalString);

echo "Reversed String: $reversedString\n";
```

?>

## Output

```
=== String Reversal using strrev() ===  
  
Original String: Hello World PHP Programming  
Reversed String: gnimmargorP PHP dlroW olleH
```

## Conclusion

The PHP program successfully reverses the input string using the `strrev()` function. The `strrev()` function is a built-in PHP string manipulation function that efficiently reverses the character order of a string. In this execution, the string 'Hello World PHP Programming' was successfully reversed to 'gnimmargorP PHP dlroW olleH'. This function is particularly useful in various string processing tasks, data validation, and text manipulation applications.