ADP LAB reference material for 21-JUNE-2023

PHP - arrays, looping over arrays, functions

```
$batch = array("N", "P", "Q"); //normal indexed array
echo $batch[0];
$asar = array("CSE"=>200, "IT"=>120,"ECE"=>120, "EEE"=>100,
"MECH"=>100); // associative array
echo $asar["CSE"]."<br>";

looparr($ar);       //function call
loopasar($asar);  //function call

function looparr($t1){          //create function with function keyword
        for ($c = 0; $c < count($t1); $c++){ // count($arr) returns array
                                                           length
        echo $t1[$c]."<br>";}}

function loopasar($t2){
        foreach ($t2 as $x => $x_value ){
        echo "Key=" . $x . ", Value=" . $x_value."<br>";}}
```

Useful array methods:

```
        In_array()
                if (in_array("N", $batch)){ echo "found";}
                else{ echo "not found";}
        array_keys()   // returns only the keys from the array
                print_r(array_keys($asar))
```

Print functions in php other than echo: print, print_r, var_dump:

```
        $batch = array("N", "P", "Q");
        print "hello". $batch[0]; //same as echo
        print_r($batch);
        echo var_dump($batch);
```

LAB EXERCISES for 21-JUNE-2023

Use php **arrays, loops, functions** for the following:

- strlen($str) function returns the length of a string.
- count($arr) returns the length of an array

1. Declare a function that accepts an array of strings and returns another array of strings arranged according to the lengths of each string, in another array.
2. Create an associative array and using a *for* loop, print the keys and their values in a HTML table (keys and their values in two columns).
3. Try print, print_r and var_dump