# Bài tập

(\*) Các form đều phải sử dụng bootstrap

(\*\*) Tất cả các bài tập đều phải yêu cầu viết hàm

(\*\*\*) Với kết quả là array, boolean, dùng print\_r($array) để in các giá trị ra

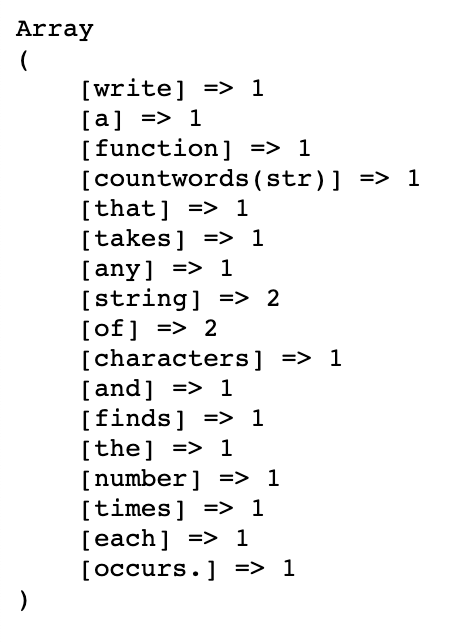
## Bài tập 1: Sử dụng hàm đơn giản trong PHP để giải quyết bài toán sau

Viết một hàm tên là countWords($str), tiếp nhận một chuỗi $str và tìm ra số lần mà mỗi từ xuất hiện trong chuỗi. Bỏ qua tất cả các trường hợp uppercase và lowercase và các kí tự đặc biệt

Ví dụ như sau:

$str = “Write a function countWords($str) that takes any string of characters and finds the number of times each string occurs.”

Kết quả



Gợi ý:

* Tạo ra một associative array có chứa các từ và số lần xuất hiện của các từ như hình ở trên. Sử dụng hàm explode($separator, $searchedString) có thể tách chuỗi thành các từ riêng biệt

explode(“ ”, “Hello there”) => [“Hello”, “there”]

Hàm này sẽ tách từ

## Bài tập 2: Sử dụng hàm đã được viết ở phần trên, viết giao diện để giải quyết bài toán sau, sử dụng form

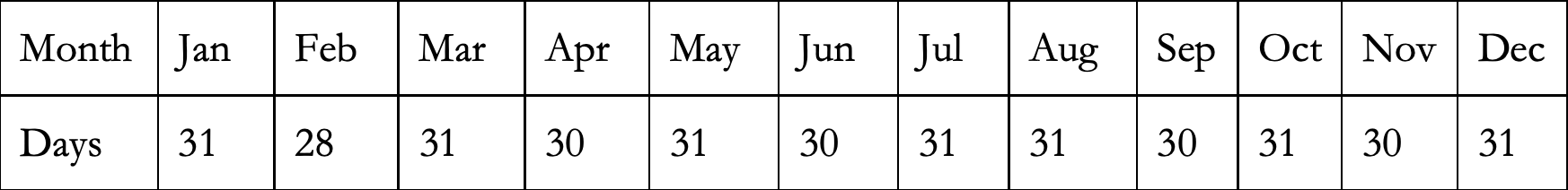
Sử dụng Bootstrap, viết một form bao gồm textfield và một button có nhiệm vụ tìm kiếm chuỗi với yêu cầu tương tự như ở trên, sử dụng chính form đó để gửi $\_SERVER[“PHP\_SELF”]

Sau khi gửi kết quả sử dụng form, hiển thị dưới dạng bảng bao gồm 2 cột

* Cột 1 chứa các từ để tìm kiếm
* Cột 2 chứa số lần xuất hiện của từ đó

## Bài tập 3: Sử dụng PHP để thực hành bài tập sau (không cần dùng form)

Viết một hàm daysInMonth() có tham số là $month và trả về số ngày trong tháng đó. Ví dụ daysInMonth(6) sẽ trả về 30



## Bài tập 4: Sử dụng PHP Form và function để thực hành bài tập sau

Viết một hàm isPalindrome() nhận một $string là tham số và cho ra kết quả chuỗi đó có phải là palindrome hay không. Một chuỗi được coi là palindrom nếu như khi viết xuôi và viết ngược đều giống nhau, ví dụ (radar, toot, mom, a, “”). Chuỗi nhập vào không cần case-sensitive: ví dụ như “RAdar” vẫn được coi là chuỗi palindrome như bình thường.

## Bài tập 5: Sử dụng PHP và form để thực hành bài tập sau

Viết một hàm reverse() để tiến hành đảo vị trí các số. Trong trường hợp số nhập vào là số âm thì vẫn phải giữ số âm đó. Nên biến đổi số thành chuỗi để thuận tiện cho phép đảo.

Ví dụ

* reverse(912) == 219
* reverse(120) === 21
* reverse(-12) === -21
* reverse(-120) === -21
* reverse('qwerty') === 'ytrewq'
* reverse('apple') === 'elppa'

Gợi ý: Sử dụng hàm is\_numeric($value)để kiểm tra xem số $value đó có phải là số hay không. Ví dụ is\_numeric(123) 🡺 TRUE

Ngoài ra strval($input) được dùng để biến đổi $input thành chuỗi “$input”

Tương tự với intval($input) được dùng để biến đổi từ chuỗi “$input” sang số $input

Cần kiểm tra cả 2 trường hợp về số nguyên bao gồm số âm và số dương

## Bài tập 6: Sử dụng PHP (không cần form) để thực hiện bài toán sau

Viết một hàm tiếp nhận một array đầu vào và một số chỉ định số phần tử được cắt trong mảng, cho phép người dùng cắt array thành các array con như sau.

* group([1, 2, 3, 4, 5], 2) -> [[ 1, 2], [3, 4], [5]]
* group([1, 2, 3, 4, 5], 3) -> [[ 1, 2, 3], [4, 5]]
* group([1, 2, 3, 4, 5], 6) -> [[ 1, 2, 3, 4, 5]]

Hướng dẫn:

* Hàm continue; có tác dụng loại bỏ các câu lệnh ở dưới hàm continue; và tiếp tục vòng lặp với giá trị tiếp theo như sau

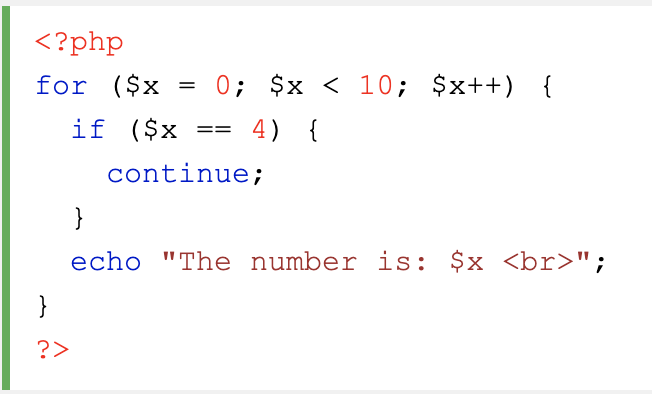


Figure : Sử dụng continue để tiếp tục vòng lặp

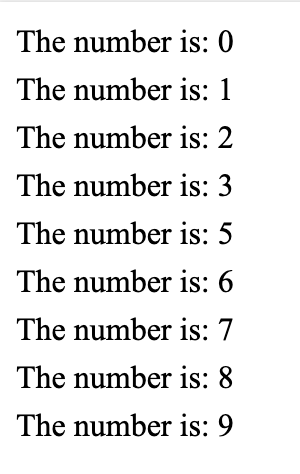


Figure : Kết qủa

Với $x = 4 gặp continue; sẽ loại bỏ toàn bộ khối lệnh ở dưới và dịch chuyển đến $x tiếp theo (=5)

## Bài tập 7: Sử dụng PHP Form, viết hàm để giải quyết bài toán sau

Không dùng hàm có sẵn, viết hàm để viết hoa chữ cái đầu tiên của các từ. Ví dụ như sau

* capitalize(“hello there”) === 'Hello There'
* capitalize("hey, so it's working!") === "Hey, So It's Working!"

Gợi ý: Sử dụng hàm ucfirst($string) để viết hoa chữ cái đầu của từng từ. Ví dụ

* ucfirst(‘hello’) 🡺 Hello

## Bài tập 8: Sử dụng PHP form, viết hàm để giải quyết bài toán sau

Xử lí bài toán Anagram: 2 chuỗi được coi là anagram nếu như tất cả chữ cái của chuỗi 1 đều được sử dụng trong chuỗi 2. Viết chương trình kiểm tra xem 2 chuỗi nhập vào có phải là angram hay không, tính cả chữ hoa chữ thường. Ví dụ như sau

* checkAnagram('rail safety', 'fairy tales') === true
* checkAnagram('roast beef', 'goat roast') === false
* checkAnagram('Elvis, 'lives') === true

Hướng dẫn: Để kiểm tra một kí tự có tồn tại ở trong chuỗi hay không, chúng ta dùng strpos($searchedString, $searchWord). Hàm này sẽ trả về chỉ mục được tìm thấy, nếu không tìm thấy $searchWord trong $searchString, false sẽ được trả về

* strpos(‘goat roast’, ‘r’) => 5
* strpos(‘goat roast’, ‘p’) 🡺 false

CHÚ Ý: Để sử dụng strpos() nếu không tồn tại, chúng ta phải sử dụng SO SÁNH 3 BẰNG (===). Nếu so sánh 2 bằng hoạt động không như mong đợi. Ví dụ so sánh 3 bằng như sau

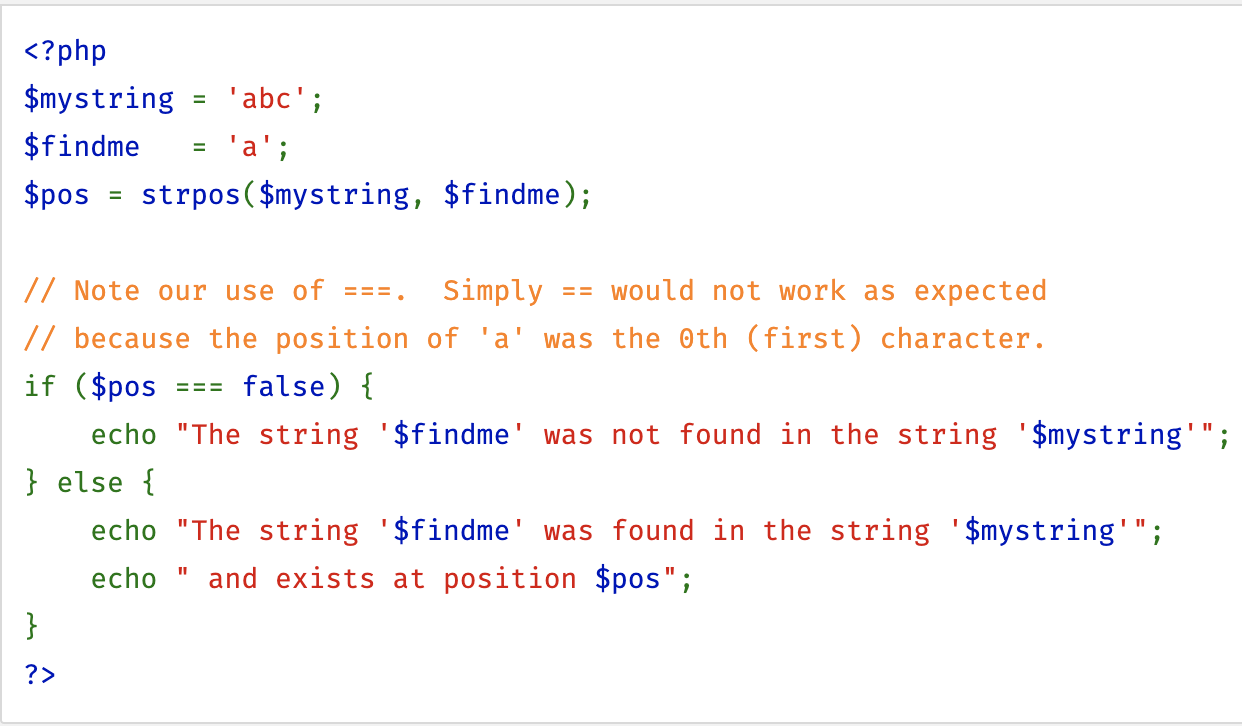


Figure : So sánh 3 bằng trong strpos()