

**ỦY BAN NHÂN DÂN THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG CAO ĐẲNG KỸ THUẬT LÝ TỰ TRỌNG TP. HCM**

-----oOo-----

**ĐỀ TÀI SÁNG KIẾN KINH NGHIỆM
NĂM HỌC 2015 - 2016**

XÂY DỰNG HỆ THỐNG BÀI TẬP LẬP TRÌNH PHP

BIÊN SOẠN ĐỀ TÀI: Trần Việt Khánh

Lời Giới Thiệu

Lập trình PHP là học phần thuộc khối kiến thức chuyên ngành đối với các sinh viên thuộc các chuyên ngành công nghệ thông tin. Lập trình PHP cũng là khối kiến thức quan trọng mà sinh viên cần phải có được để trang bị hành trang làm việc sau khi tốt nghiệp đáp ứng nhu cầu trong thị trường công nghệ thông tin ngày nay.

Hệ thống bài tập lập trình PHP này trình bày các chủ đề bài tập liên quan đến ngôn ngữ lập trình PHP và cơ sở dữ liệu MySQL. Mỗi chủ đề được thiết kế bao gồm các phần: Tóm tắt lý thuyết, hướng dẫn giải một số dạng bài tập cơ bản và một số bài tập chọn lọc nâng cao. Trong mỗi chủ đề đều có các ví dụ minh họa và mỗi ví dụ đều có trình bày về code mẫu, kết quả đạt được khi chạy ứng dụng Web. Phần cuối của hệ thống bài tập lập trình PHP có hướng dẫn giải cho một số bài tập tiêu biểu, đồng thời bổ sung một số bài tập tổng hợp, nâng cao để sinh viên tham khảo và tự rèn luyện kỹ năng phân tích vấn đề, rèn luyện tư duy và phong cách lập trình.

Hệ thống bài tập lập trình PHP này được biên soạn để giảng dạy cho học phần lập trình PHP ở bậc cao đẳng. Hệ thống bài tập này cũng có thể làm tài liệu tham khảo cho các giáo viên đang giảng dạy kiến thức lập trình PHP ở các bậc đào tạo khác, và giúp cho sinh viên tự học có kết quả. Tùy theo nội dung chương trình đào tạo mà giáo viên có thể chọn một số nội dung phù hợp trong hệ thống bài tập này để giảng dạy.

Tôi trân trọng giới thiệu với bạn đọc hệ thống bài tập lập trình PHP này và hy vọng rằng nó sẽ giúp cho việc giảng dạy và học tập môn lập trình PHP được thuận lợi hơn.

Cuối cùng, chúng tôi xin gửi lời cảm ơn chân thành đến các đồng nghiệp trong khoa công nghệ thông tin trường Cao Đẳng Kỹ Thuật Lý Tự Trọng TP. HCM đã có nhiều đóng góp quý báu, đã cùng tôi chia sẻ nội dung tập bài giảng lập trình PHP này trong suốt thời gian qua.

Thành phố Hồ Chí Minh, ngày 11 tháng 11 năm 2015

Trần Việt Khánh

Chương 1

CÁC KHÁI NIỆM PHP CƠ BẢN

A. TÓM TẮT LÝ THUYẾT

1.1. Giới thiệu

- ✓ PHP với tên gốc là Personal Home Page. Là ngôn ngữ mã nguồn mở để viết các trang web động.
- ✓ Apache: máy chủ web để chạy các ứng dụng PHP sau khi lập trình.
- ✓ Kết hợp với hệ quản trị CSDL MySQL

1.2. Các thành phần cơ bản PHP

- ✓ Nhúng PHP vào HTML, có thể nhúng mã PHP vào mọi vị trí trong trang HTML.
- ✓ Các đoạn mã PHP phải được nằm trong thẻ mở `<?php` và thẻ đóng `?>`

`<?php`

`// Đoạn lệnh PHP ở đây`

`?>`.

- ✓ Một cấu trúc lệnh thông thường của PHP có thể được tách làm nhiều phần, mỗi phần đặt giữa `<?php ... ?>`
- ✓ Đặc điểm của PHP là có khả năng hướng đối tượng, phân biệt chữ hoa/chữ thường, mỗi lệnh kết thúc bởi chấm phẩy (;)
- ✓ Chú thích trong PHP, chú thích một dòng lệnh (// Dòng chú thích), chú thích trên một đoạn lệnh (/* Đoạn chú thích */)

1.3. Biến, hằng, các toán tử

✓ Biến

- Bắt đầu bằng dấu đô la (\$), tiếp ngay sau \$ là tên biến.
- Tên biến bao gồm chữ cái, chữ số, dấu gạch nối (_) và phải bắt đầu bằng chữ cái hoặc dấu gạch nối.
- Ví dụ: \$a, \$b1, \$_c,

✓ Hằng

- Hằng cũng là một biến nhưng giá trị của nó không thể thay đổi. Tuy nhiên cách khai báo biến và hằng số thì lại khác nhau.
- **Cú Pháp:** `define('ten_hang', 'gia_tri');`

Ví dụ: Tạo một hằng số có tên là SDT và gán giá trị cho nó là 0909090909

```
define('SDT', '0909090909');
```

✓ Các toán tử và kiểu dữ liệu

- **Toán tử gán**: gán giá trị cho 1 biến,
Ví dụ: `$number = 10 ; // gán giá trị 10 cho biến $number`
- **Toán tử số học**: các phép toán cơ bản cộng, trừ, nhân, chia, ...

Tên	Ký hiệu	Mô tả	Ví dụ
Phép cộng	+	Cộng hai số hạng	$\$a + \b
Phép trừ	-	Trừ hai số hạng	$\$a - \b
Phép nhân	*	Nhân hai số hạng	$\$a * b$
Phép chia	/	Chia hai số hạng	$\$a / \b
Phép chia lấy dư	%	Chi lấy dư	$5 \% 2 = 1$

Ví dụ :

```
$a = 5;
$b = 10;
$c = $a + $b;
$d = $a * $b;
```

- **Toán tử so sánh**: các phép toán logic như so sánh bằng, lớn hơn, nhỏ hơn,...

Tên	Ký hiệu	Mô tả	Ví dụ
So sánh bằng	==	Hai số hạng bằng nhau	$\$a == \b
So sánh khác	!=	Hai số hạng khác nhau	$\$a != \b
So sánh lớn hơn	>	So sánh lớn hơn	$\$a > b$
So sánh nhỏ hơn	<	So sánh nhỏ hơn	$\$a < \b
Lớn hơn hoặc bằng	>=	Lớn hơn hoặc bằng	$\$a >= \b
Nhỏ hơn hoặc bằng	<=	Nhỏ hơn hoặc bằng	$\$a <= \b

Ví dụ :

```
$a = 5;
$b = 10;
if($a > $b){
    echo "$a lớn hơn $b";
}else{
    echo "$a nhỏ hơn $b";
}
```

- **Toán tử logic**: Toán tử logic là tổ hợp các giá trị boolean có kết quả trả về là TRUE hoặc FALSE

Tên	Ký hiệu	Mô tả	Ví dụ
Phép And	&&	Cả 2 vế phải thoả mãn điều kiện	$\$a > \$b \&\& \$a > \c
Phép Or		Một trong 2 vế thoả mãn điều kiện là được	$\$a > \$b \$a < \c
Phủ định	!	Phủ định 1 điều kiện, giá trị nào đó	$\$a != b$

Ví dụ :

```

$diem = 7;
if($diem <= 5) {
    echo "Bạn học dốt quá!";
}elseif($diem > 5 && $diem <= 8) {
    echo "Bạn học rất khá!";
}else{
    echo "Bạn học rất giỏi!";
}

```

- **PHP hỗ trợ 4 kiểu dữ liệu**

- Kiểu số
 - + Số nguyên từ -2^{31} đến $2^{31}-1$
 - ❖ Hệ thập phân: \$a = 16;
 - ❖ Hệ 16 (hexa): \$a=0x10;
 - ❖ Hệ 8 (bát phân): 020;
 - + Số thực (thập phân): từ 1.7E-308 đến 1.7E+308
 - ❖ Biểu diễn: \$a=0.017;
 - ❖ Dạng khoa học: \$a=17.0E-03;
- Kiểu chuỗi
 - + Giới hạn bởi nháy đơn (‘) hoặc kép (“)
 - + Chuỗi đặt trong nháy kép bị thay thế, trong nháy đơn thì không.
 Ví dụ:


```

$a= "Hello";
$b= "$a world"; //tương đương $b="Hello world"
$c= ' $a world'; //$c=' $a world' (không thay đổi)

```
- Kiểu logic
 - + Có 2 trạng thái: true và false
 Ví dụ:


```

$x = true;
$y = false;

```

1.4. Các cấu trúc điều khiển.

1.4.1. Cấu trúc rẽ nhánh

Dạng 1

```

if (<điều kiện>)
    <công việc>;

```

Mô tả hoạt động

Nếu <điều kiện> là đúng thì thực hiện <công việc>, nếu <điều kiện> là sai thì sẽ không thực hiện <công việc>.

Lưu ý: Nếu <công việc> có từ hai câu lệnh đơn trở lên thì cần đặt <công việc> giữa hai dấu { }.

Dạng 2

```

if (<điều kiện>)
    <công việc 1>;

```

else

<công việc 2>;

Mô tả hoạt động

Nếu <điều kiện> là đúng thì sẽ thực hiện <công việc 1>, ngược lại nếu <điều kiện> là sai thì <công việc 2> sẽ được thực hiện.

1.4.2.Cấu trúc lặp với số lần lặp biết trước

for (<biểu thức 1>; <điều kiện>; <biểu thức2>)

<công việc>;

Mô tả hoạt động

Đầu tiên sẽ thực hiện <biểu thức 1>, sau đó kiểm tra <điều kiện>. Nếu <điều kiện> là đúng thì sẽ thực hiện <công việc> rồi đến <biểu thức 2>; sau khi thực hiện xong <biểu thức 2> thì cấu trúc này quay trở lại kiểm tra <điều kiện> và bắt đầu một vòng lặp mới,...cứ thế vậy đến khi <điều kiện> nhận giá trị sai thì cấu trúc lặp này sẽ kết thúc.

1.4.3.Sơ đồ lặp với số lần lặp không biết trước

Dạng 1

while (<điều kiện>)

<công việc>;

Mô tả hoạt động

Nếu <điều kiện> là đúng thì sẽ thực hiện <công việc>, sau đó quay trở lại thực hiện <điều kiện>. Vòng lặp while này sẽ kết thúc khi <điều kiện> là sai.

Dạng 2

do

{

<công việc>;

}

while (<điều kiện>);

Mô tả hoạt động

Trước hết <công việc> sẽ được thực hiện, sau đó kiểm tra <điều kiện>. Nếu <điều kiện> là đúng thì quay trở lại thực hiện <công việc> và bắt đầu một vòng lặp mới. Vòng lặp do while này sẽ kết thúc khi <điều kiện> nhận giá trị sai.

1.5.Hàm

✓ Khai báo hàm

- `function tên_hàm(ts1, ts2,..., tsn){`
 `//các lệnh trong thân hàm`
 `}`
- Để trả ra giá trị cho hàm ta sử dụng lệnh **return** biểu_thức;
- Để thoát khỏi hàm: Sử dụng `return` hoặc `exit`
- Ví dụ: Xây dựng hàm tính tổng 2 số
 `function tinh tong($a, $b)`
 `{`
 `return $a + $b;`
 `}`

✓ Gọi hàm

- `tên_hàm(gt1, gt2,..., gtn);`
Ví dụ: gọi hàm tính tổng 2 số
 `$so1 = 12;`
 `$so2 = 13;`
 `echo tinh tong($so1, $so2);`
- **Truyền tham biến và tham trị.**
Việc gọi hàm tính tổng (`tinh tong($so1,$so2)`), tất cả việc truyền biến đều là tham trị. Để truyền tham biến (biến được truyền có thể thay đổi nội dung khi ra khỏi hàm), khi gọi hàm, ta đặt thêm dấu `&` trước biến cần truyền tham biến.
- Ví dụ về truyền tham biến

`<?php`

```
function F($a, $b){
    $a = 5;
    $b = 5;
    echo "<br> Trong hàm F(), a = ".$a." và b = ".$b;
}
$a = 8;
$b = 8;
echo "<br> 1. Ngoài hàm F(), a = ".$a." và b = ".$b;

//Gọi hàm tham trị.
F($a, $b);
echo "<br> 2. Ngoài hàm F(), a = ".$a." và b = ".$b;

//Gọi hàm với biến $a là tham biến.
F(&$a, $b);
```

echo "
3. Ngoài hàm F()a = ".\$a." và b = ".\$b;

?>

Kết quả:

1. Ngoài hàm F(), a = 8 và b = 8

Trong hàm F(), a = 5 và b = 5

2. Ngoài hàm F(), a = 8 và b = 8

Trong hàm F(), a = 5 và b = 5

3. Ngoài hàm F(), a = 5 và b = 8

1.6 Mảng và chuỗi

+ Khởi tạo mảng một chiều với dữ liệu là chuỗi bằng hàm array()

❖ Chỉ số mặc định với chỉ số đầu bằng 0

```
$a=array("xin", "chào", "bạn");
```

```
$a[0]="xin";
```

```
$a[1]="chào";
```

```
$a[2]="bạn";
```

❖ Chỉ số mặc định với chỉ số đầu khác 0

```
$a=array(5=>"xin", "chào", "bạn");
```

```
$a[5]="xin";
```

```
$a[6]="chào";
```

```
$a[7]="bạn";
```

+ Khởi tạo mảng nhiều chiều với dữ liệu là chuỗi bằng hàm array()

```
$monhoc = array(array("HTML4","HTML5"),
                 array("CSS2","CSS3"),
                 array("Jquery","Jquery Mobile"));
```

```
$monhoc[0][0]= "HTML4"
```

```
$monhoc[0][1]= "HTML5"
```

```
$monhoc[1][0]= "CSS2"
```

```
$monhoc[1][1]= "CSS3"
```

```
$monhoc[2][0]= "Jquery"
```

```
$monhoc[2][1]= "Jquery Mobile"
```

1.7.Tập tin

✓ File được tải lên và lưu trên máy chủ có tên là \$_FILES["UserFileName"] với các thông tin như sau:

- \$_FILES["UserFileName"]['name']: Tên gốc của file trên máy khách

- \$_FILES["UserFileName"]['type']: Kiểu MIME của file. Ví dụ: "image/gif", "image/jpeg",.....

- \$_FILES["UserFileName"]['size']: Kích thước của file được tải tính theo byte

- \$_FILES["UserFileName"]['tmp_name']: Tên tạm của file khi nó đã được tải và lưu trên thư mục tạm của máy chủ.

✓ Để **upload được file**, chúng ta cần thêm thuộc tính **enctype="multipart/form-data"** cho form, ta dùng thẻ input type="file" để **upload file**

- Tạo file form.php với nội dung như sau

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Upload file trong PHP</title>
</head>

<body>
<h3>PHPPANDMYSQL.NET</h3>
<form action="form.php" method="post" enctype="multipart/form-data">
    Select file : <input type="file" name="file" size="20" /><br />
    <input type="submit" value="Upload" name="ok" />
</form>
</body>
</html>
```

- Tiếp theo ta sẽ kiểm tra xem người dùng đã nhấn nút submit hay chưa bằng cách kiểm tra sự tồn tại của biến `$_POST['ok']`, nếu người dùng đã nhấn submit rồi thì tiến hành kiểm tra xem họ đã chọn file hay chưa, nếu chưa chọn file thì báo lỗi

- Ta có thể lấy được tên file, kiểu file, kích cỡ file thông qua các biến :

```
Tên file : $_FILES['file']['name'];
Kiểu file : $_FILES['file']['type'];
Kích thước file : $_FILES['file']['size'];
<?php
    if(isset($_POST['ok'])) { // Người dùng đã ấn submit
        if($_FILES['file']['name'] != NULL) { // Đã chọn file
            // thực hiện công việc upload
        } else {
            echo "Vui lòng chọn file";
        }
    }
?>
```

- Khi người dùng đã chọn file thì tiến hành upload ? tùy theo yêu cầu của bài toán mà chúng ta tiến hành xây dựng chức năng cho nó, ví dụ yêu cầu đặt ra ở đây là chỉ được upload các file ảnh có kiểu file là jpg, png, gif và cỡ file không được lớn hơn 1mb

```
<?php
    if ( $_FILES['file']['type'] == "image/jpeg"
    || $_FILES['file']['type'] == "image/png"
    || $_FILES['file']['type'] == "image/gif") { // là file ảnh
        // Tiến hành code upload
    } else {
        // không phải file ảnh
        echo "Kiểu file không hợp lệ";
    }
?>
```

- Bước cuối cùng là xem file vừa chọn kích cỡ có nhỏ hơn 1mb không, file size ở đây được tính theo byte (1mb = 1*1024 kb * 1024 bytes). Nếu cỡ file nhỏ hơn 1mb tiến

hành upload file bằng hàm move_uploaded_file , file upload sẽ được lưu trong thư mục images

```
<?php
if($_FILES['file']['size'] > 1048576){
    echo "File không được lớn hơn 1mb";
}else{
    // file hợp lệ, tiến hành upload
    $path = "images/"; // file lưu vào thư mục images
    $tmp_name = $_FILES['file']['tmp_name'];
    $name = $_FILES['file']['name'];
    $type = $_FILES['file']['type'];
    $size = $_FILES['file']['size'];
    // Upload file
    move_uploaded_file($tmp_name,$path.$name);
    echo "File uploaded! <br />";
    echo "Tên file : ".$name."<br />";
    echo "Kiểu file : ".$type."<br />";
    echo "File size : ".$size;
}
?>
```

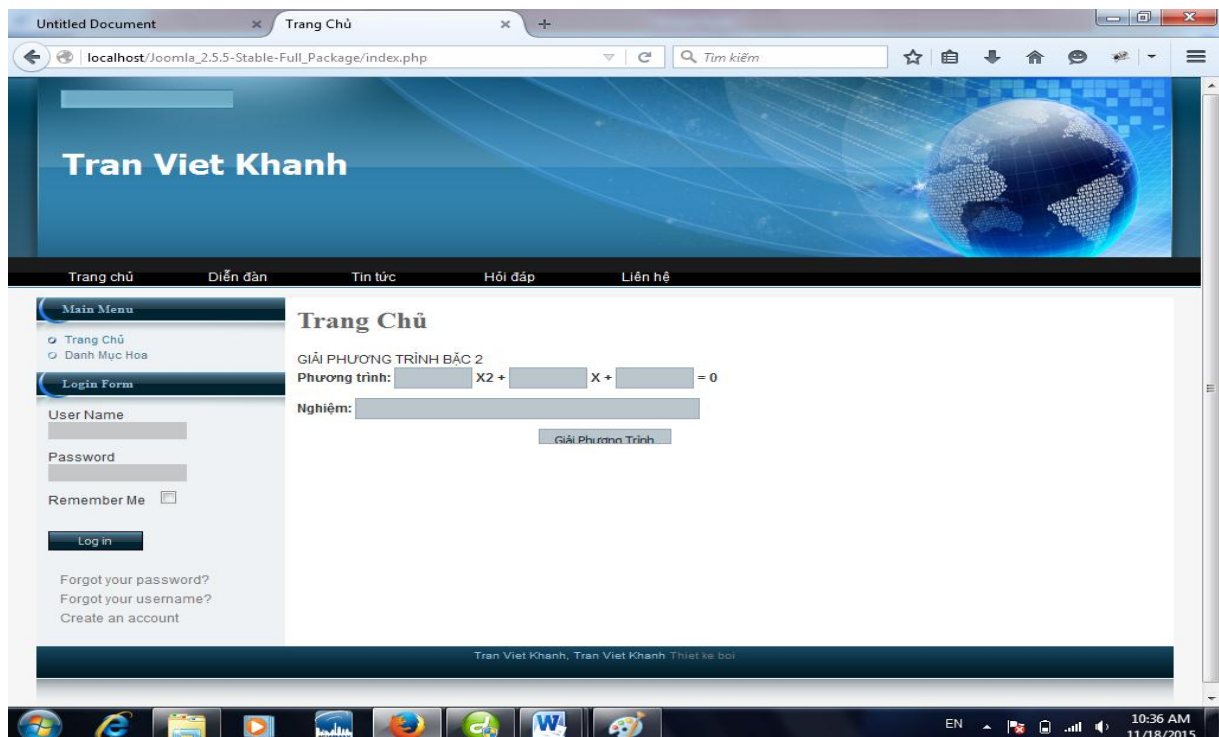
1.8.Tạo trang PHP đầu tiên.

Trong học phần lập trình PHP này, khi viết xong một chương trình ứng dụng web nào đó thì sinh viên cần kiểm thử lại chương trình qua một số bộ test mẫu; để từ đó có thể phát hiện sai sót hoặc có thể cải tiến để được một chương trình hiệu quả hơn về mặt thời gian hoặc không gian lưu trữ dữ liệu,...

B.CÁC DẠNG BÀI TẬP

❖ Dạng 1: Cấu trúc rẽ nhánh

Bài tập 1-1. Xây dựng ứng dụng Web giải phương trình dạng $ax^2 + bx + c = 0$ như hình sau đây:



Hướng dẫn giải bài tập 1-1: Sinh viên viết chương trình theo code mẫu sau đây:

<?php

```
$nghiem=NULL;
function GiaiPhuongTrinhBacHai($a,$b,$c)
{
    if($a==0)
    {
        if($b==0)
        {
            if($c==0)
                $nghiem = "Phương trình có vô số nghiệm";
            if($c!=0)
                $nghiem = "Phương trình vô nghiệm";
        }
        else if($b!=0)
        {
            $x=(-$c/$b);
            $x= round($x,2);
            $nghiem = "Phương trình có nghiệm: x = $x";
        }
    }
    if($a!=0)
    {
        $delta = pow($b,2)- 4*$a*$c;
        if($delta<0)
            $nghiem = "Phương trình vô nghiệm";
        if($delta==0)
        {
            $nghiemkep = (-$b/2*$a);
            $nghiem = "Phương trình có nghiệm số kép x1 = x2 = $nghiemkep";
        }
        if($delta>0)
        {
            @$x1 = (-$b + sqrt($delta))/(2*$a);
            @$x2 = (-$b - sqrt($delta))/(2*$a);
            $nghiem = "Phương trình có hai nghiệm phân biệt x1=".round($x1,2).",
                                                                x2=".round($x2,2);
        }
    }
    return $nghiem;
}
if(isset($_POST["a"]) && isset($_POST["b"]) && isset($_POST["c"]))
{
    $a=$_POST["a"];
    $b=$_POST["b"];
    $c=$_POST["c"];
    $nghiem=GiaiPhuongTrinhBacHai($a,$b,$c);
}
```

?>

```
<form action="<?php echo $_SERVER['PHP_SELF'];?>" method="post">
    <fieldset><legend>GIẢI PHƯƠNG TRÌNH BẬC 2</legend>
    <p><b>Phương trình:</b>
        <input type="text" name="a" size="5" maxlength="5" value="<?php if(isset($_POST['a']))
        echo $_POST['a'];?>" />
    <b> X2 +</b>
```

```

<input type="text" name="b" size="5" maxlength="5" value="<?php if(isset($_POST['b']))
echo $_POST['b'];?>" />
<b> X </b>
<input type="text" name="c" size="5" maxlength="5" value="<?php if(isset($_POST['c']))
echo $_POST['c'];?>" />
<b> = 0</b>
</p>
<p><b>Nghiem:</b>
<input type="text" name="nghiem" size="50" maxlength="50" value="<?php echo
$ngkiem;?>" />
</p>
</fieldset>
<div align="center"><input type="submit" name="submit" value="Giải Phương Trình" /> </div>
</form>

```

➤ **Một số kết quả khi chạy chương trình:**

Số lần test	Nhập số a	Nhập số b	Nhập số c	Kết quả
test 1	0	0	0	Phương trình vô số nghiệm
test 2	0	0	1	Phương trình vô nghiệm
test 3	0	2	5	Phương trình có một nghiệm $x=-2.5$
test 4	1	3	4	Phương trình vô nghiệm
test 5	1	-4	4	Phương trình có nghiệm kép $x_1=x_2=2$
test 6	1	4	3	Phương trình có hai nghiệm $x_1=-1, x_2=-3$

❖ **Dạng 2: Cấu trúc lặp với số lần lặp biết trước**

Bài tập 1-2. Xây dựng ứng dụng Web tính tổng $S(n) = 1 + 2 + \dots + n$ với n là số tự nhiên ≥ 0 như hình sau đây:



Hướng dẫn giải bài tập 1-2: Sinh viên viết chương trình theo code mẫu sau đây:

<?php

```
error_reporting("E_ERROR");
$tong=NULL;
function Sn($n)
{
    for($i=0;$i<=$n;$i++)
        $tong=$tong+$i;
    return $tong;
}
if(isset($_POST["n"]))
{
    $n=$_POST["n"];
    $tong=Sn($n);
}
```

?>

```
<form action="<?php echo $_SERVER['PHP_SELF'];?>" method="post">
    <fieldset><legend><font color="#FF0000" size="+1">Tính tổng S(n) = 1 + 2 + ... + n với n là số tự
    nhiên >=0</font></legend>
    <p><b>Nhập n=:</b><input type="text" name="n" size="5" maxlength="5" value="<?php
    if(isset($_POST['n'])) echo $_POST['n'];?>" /></p>
```

<p>S(n):

<input type="text" name="tong" size="50" maxlength="50" value="<?php echo \$tong;?>" /></p>

</fieldset>

<div align="center"><input type="submit" name="submit" value="Tính tổng" /> </div>

</form>

➤ **Một số kết quả khi chạy chương trình:**

Số lần test	Nhập N	S(N)
test 1	0	0
test 2	1	1
test 3	10	55
test 4	100	5050