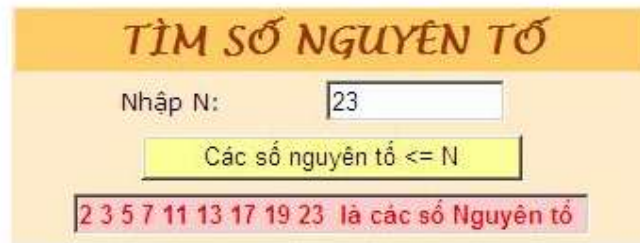


25. Tìm số nguyên tố



TÌM SỐ NGUYÊN TỐ

Nhập N:

Các số nguyên tố <= N

2 3 5 7 11 13 17 19 23 là các số Nguyên tố

Hướng dẫn giải:

– Lấy giá trị N trên form thông qua biến \$_POST

– Viết hàm kiểm tra số nguyên tố

```
function kt_snt($so)
```

```
{
```

Hàm này có kết quả trả về là 1 nếu số được truyền vào là số nguyên tố

Ngược lại thì trả về 0

```
}
```

– Tìm các số nguyên tố <= N (dùng vòng lặp for)

○ Nếu $N < 2$: thì không có SNT (dùng if)

○ Ngược lại thì (dùng else else)

```
{
```

Tạo một chuỗi có giá trị là 2 trước (vì 2 là SNT)

Dùng vòng lặp duyệt từ 3 đến N (for \$i)

Gọi hàm kt_snt(\$i), kiểm tra nếu kết quả trả về là 1 thì ghi SNT này vào chuỗi

```
}
```

– Xuất ra các số nguyên tố

26. Tìm USCLN và BSCNN

ƯỚC SỐ CHUNG LỚN NHẤT và BỘI SỐ CHUNG NHỎ NHẤT	
Số A:	<input type="text" value="45"/>
Số B:	<input type="text" value="60"/>
USCLN:	<input type="text" value="15"/>
BSCNN:	<input type="text" value="180"/>
<input type="button" value="Tìm USCLN và BSCNN"/>	

Yêu cầu: Khi chọn, thực hiện :

1. Tìm ước số chung lớn nhất của 2 số A và B
2. Tìm bội số chung nhỏ nhất của 2 số A và B
3. Xuất kết quả ra TextField USCLN và BSCNN

Ghi chú:

USCLN của 2 số A và B là số lớn nhất mà cả 2 cùng chia hết cho số đó

$BSCNN = (số A * số B) / USCLN$

Hướng dẫn giải:

- Lấy giá trị số A, số B trên form thông qua biến `$_POST`
- Dùng vòng lặp do...while để tìm USCLN (`$uscln`)
- Tìm BSCNN bằng công thức: $BSCNN = (số A * số B) / USCLN$
`$bscnn = ($a*$b)/$uscln;`
- Xuất `$uscln` và `$bscnn` vào TextField USCLN và BSCNN

27. Giải phương trình bậc 2

GIẢI PHƯƠNG TRÌNH BẬC HAI

Phương trình: x^2 + x + = 0

Nghiệm:

Yêu cầu:

- Khi chọn, thực hiện giải phương trình bậc hai theo thuật toán sau:

Nếu $a=0$:

§ Giải phương trình bậc nhất: $bx + c = 0$

Ngược lại (nếu $a \neq 0$)

§ Tính $\Delta = b^2 - 4ac$

§ Tìm nghiệm theo Δ

- $\Delta < 0$: Phương trình vô nghiệm
- $\Delta = 0$: Phương trình có nghiệm kép $x_1 = x_2 = -b / 2a$
- $\Delta > 0$: Phương trình có 2 nghiệm:

$$x_{1,2} = \frac{-b \pm \sqrt{\Delta}}{2a}$$

Hướng dẫn giải:

- Lấy giá trị của a, b, c trên form thông qua biến $$_POST$
- Viết hàm `Giai_pt_bac_nhat`
- Sau đó viết hàm `Giai_pt_bac_hai` và gọi sử dụng hàm `Giai_pt_bac_nhat`
- In kết quả

```
function giai_ptb1($a,$b)
```

```
{
```

Lần lượt xét các trường hợp của pt bậc nhất (dùng cấu trúc if)

- Pt có vô số nghiệm

- Phương trình vô nghiệm

- Pt có nghiệm

```
}
```

```
function giai_ptb2($a, $b, $c)
```

```
{
```

Nếu $a=0$ (dùng if)

Gọi hàm *giai_ptb1*(\$b,\$c);

Nếu $a \neq 0$ (dùng if)

{

- o Tính delta
- o Nếu $\Delta < 0$: Phương trình vô nghiệm
- o Nếu $\Delta = 0$: Phương trình có nghiệm kép
- o Nếu $\Delta > 0$: Phương trình có hai nghiệm phân biệt

}

Trả về nghiệm của phương trình

}

- Gọi hàm *giai_ptb2* => Tìm và xuất nghiệm của phương trình
- \$nghiem = *giai_ptb2*(\$a, \$b, \$c)

28. Đọc số có ba chữ số



Hướng dẫn:

- Lấy giá trị số trên form thông qua biến \$_POST
- Viết hàm đọc một số sau đó gọi hàm này để đọc một số có từ 1 đến 3 chữ số, sử dụng switch ... case để đọc 1 số

```
function doc_1_so($so)
{
    o Dùng switch ... case
    o Trả về dạng chữ của một số
}
```
- Lần lượt tách số ra thành 3 phần: đơn vị - chục – trăm
- Trường hợp cả 3 phần này đều bằng 0 thì xuất ra “Không”
- Ngược lại, xét các trường hợp sau:
 - o Nếu phần đơn vị khác 0 và khác 5 thì đơn vị = doc_1_so(\$dv)
 - o Nếu phần đơn vị = 5 và phần chục <> 0 thì đơn vị = “Lăm”
 - o Nếu phần đơn vị = 5 và chục =0 thì đơn vị = “Năm”
 - o Nếu phần đơn vị = 1 và chục >1 thì đơn vị = “Mốt”
 - o Nếu phần đơn vị = 1 và chục <=1 thì đơn vị = “Một”
 - o Ngược lại (đơn vị = 0 và chục = 0) đơn vị = “”
 - o Nếu chục khác 0 và khác 1 thì chục = doc_1_so(\$chuc) . “ Mươi”;
 - o Nếu chục = 1 thì chục = “Mười” ngược lại nếu (trăm khác 0 và chục = 0) thì chục = “Lẻ”
 - o Nếu trăm khác 0 thì trăm = doc_1_so(\$tram) . “ Trăm”;
 - o Ngược lại, trăm = “”
- Ghép 3 phần lại tạo thành chuỗi đọc số như sau:

```
$chu = $doc_tram . “ ”. $doc_chuc . “ ”. $doc_dv;
```
- Xuất \$chu vào TextField Đọc số

