

Q1A	
Q1B	
Q1C	
Q2	
Q3	

Fundação CECIERJ – Vice Presidência de Educação Superior à Distância
Curso de Tecnologia em Sistemas de Computação
Disciplina: Programação II
AP3 – 1º Semestre de 2016

Nome: _____

Questão 1:

Suponha nesta questão que você já está conectado ao banco de dados MySQL que contém as seguintes tabelas:

Pessoa
id_pessoa
nome
cpf
data_nascimento
sexo
email

Amizade
id_pessoa1
id_pessoa2

*** suponha que A ser amigo de B é equivalente a B ser amigo de A. Desta for amizades equivalentes só devem estar armazenadas uma única vez.**

A) Escreva uma consulta para listar todas as mulheres registradas na base de dados em ordem crescente por idade (da mais nova para a mais velha).

Select * from Pessoa where sexo='f' order by data_nascimento DESC;

B) Escreva um código PHP/MySQL para descobrir a pessoa com mais amigos registrados no banco de dados.

```
//suponha que ja está conectado
```

```
$sql = "SELECT id_pessoa FROM PESSOA ;";
```

```
$result = $conn->query($sql);
```

```
if(!$result || $result->num_rows < 1){
```

```
    echo "Nao há pessoas cadastradas <br>";
```

```
    return -1;
```

```
}
```

```
$mais_amigos = 0;
```

```
$pessoa_mais_amigos = 0;
```

```
$rr = $result->fetch_all();
```

```
foreach ($rr as $row) {
```

```
    $id_pessoa = $row[0];
```

```
    $sql = "SELECT * FROM AMIZADE WHERE  
id_pessoa1=$id_pessoa OR id_pessoa2=$id_pessoa;";
```

```
    $result2 = $conn->query($sql);
```

```
    $num_amigos = 0;
```

```
    if(!$result2 || $result2->num_rows < 1){
```

```
        echo "Nao ha resultados a exibir <br>";
```

```
        return -1;
```

```
    }else{
```

```
        $num_amigos = $result2->num_rows;
```

```
    }
```

```
    if( $num_amigos>$mais_amigos){
```

```
        $mais_amigos = $num_amigos;
```

```
        $pessoa_mais_amigos = $id_pessoa;
```

```
}  
}
```

```
echo "Pessoa com mais amigos foi: $pessoa_mais_amigos<br>";
```

C) Escreva um código PHP/MySQL para inserir a amizade entre "João" (cpf:123456789) e "Maria" (cpf:987654321).

Você deve verificar se ambos já foram previamente cadastrados, e em caso contrário emitir uma mensagem de alerta e não cadastrar a amizade.

Deve também verificar se essa amizade já foi previamente cadastrada, emitindo uma mensagem de erro neste caso.

```
$sql = "SELECT id_pessoa FROM Pessoa WHERE NOME='João' AND  
cpf ='123456789'";
```

```
$result = $conn->query($sql);  
if(!$result || $result->num_rows < 1){  
    echo "Joao não está cadastrado <br>";  
    return -1;  
}
```

```
$row = $result->fetch_all();  
$idJoao = $row[0][0];
```

```
$sql = "SELECT id_pessoa FROM Pessoa WHERE NOME='Maria'  
AND cpf ='987654321'";
```

```
$result = $conn->query($sql);
```

```
if(!$result) || $result->num_rows < 1){  
    echo "Maria não está cadastrada <br>";  
    return -1;  
}
```

```
$row = $result->fetch_all();  
$idMaria = $row[0][0];
```

```
$sql = "SELECT * FROM AMIZADE WHERE (id_pessoa1=$idJoao  
AND id_pessoa2=$idMaria) OR (id_pessoa2=$idJoao and  
id_pessoa1=$idMaria);";
```

```
$result = $conn->query($sql);  
if(!$result) || $result->num_rows < 1){
```

```
    $sql = "INSERT INTO AMIZADE (id_pessoa1,id_pessoa2)  
values ($idJoao,$idMaria);";
```

```
    $result = $conn->query($sql);
```

```
}
```

```
else{
```

```
    echo "Erro:essa amizade já foi previamente  
cadastrada<BR>" ;
```

```
}
```

Questão 2:

Que diferenças existem entre envio de dados por GET e por POST? Descreva sobre quando é preferível usar um ou outro e como as informações são enviadas por um e por outro.

A maior diferença entre os métodos GET e POST é a visibilidade: uma requisição GET é enviada como string acrescentado seu conteúdo a URL, enquanto que para POST é encapsulada junto ao corpo da requisição HTTP e não pode ser vista.

Outra diferença é o tamanho permitido a mensagem enviada. Já que com GET a mensagem é enviada via URL, seu tamanho é limitado, enquanto que usando POST não há limitações de comprimento, já que a mesma é enviada no corpo da requisição HTTP.

Sobre a escolha entre os dois, é preferível usar GET quando a mensagem é pequena e pode ser revelada por ser relativamente mais rápida. Por outro lado, é preferível usar post quando queremos preservar oculto seu conteúdo, quando a mensagem contém outras informações que não textos (GET só aceita textos e limitados ao padrão ASCII), ou ainda quando for grande.

Questão 3:

Escreva um código PHP para normalizar um vetor de tamanho qualquer contendo apenas números (ou seja, ao final da execução os valores do vetor devem estar entre o intervalo [0.0,1.0]). Exemplos:

Entrada: array(1,2,3)	Entrada: array(1,2,3,4,5)	Entrada: array(100,150,200)
Saída: array(0, 0.5, 1)	Saída: array(0, 0.25, 0.5, 0.75, 1)	Saída: array(0, 0.5, 1.0)

```
function normalizar(&$v){

    $max =$v[0];
    $min = $v[0];
    foreach( $v as $value ){

        if( $value > $max)
            $max = $value ;
        if( $value < $min)
            $min = $value ;

    }

    for($i = 0; $i < count($v); $i++){

        $v[$i] = ($v[$i]-$min)/($max-$min);
    }

    return $v;
}
```