PROCERGS

CENTRO DE SOLUÇÕES EM GOVERNO ELETRÔNICO

CONCURSO PÚBLICO PARA PROVIMENTOS DE VAGAS EM CARGOS DE NÍVEL SUPERIOR E MÉDIO

TÉCNICO DE COMPUTAÇÃO - TC
PROGRAMADOR PHP

INSTRUÇÕES

Leia atentamente e cumpra rigorosamente as instruções que seguem, pois elas são parte integrante das provas e das normas que regem este Concurso Público.

1. Verifique se o cargo constante na capa deste caderno é aquele para o qual realizou a inscrição.
2. Cada questão oferece 5 (cinco) alternativas de respostas, representadas pelas letras **A**, **B**, **C**, **D** e **E**, sendo apenas 1 (uma) a resposta correta.
3. O tempo para a realização da prova é de 4 horas, incluindo o preenchimento da grade de respostas. O candidato só poderá retirar-se do recinto da prova teórico-objetiva após transcorrida 1 hora e 30 minutos de seu início. Os dois últimos candidatos deverão retirar-se da sala de prova ao mesmo tempo, devendo assinar a Ata de Prova.
4. Nenhuma informação sobre as instruções e/ou sobre o conteúdo das questões será dada pelo fiscal, pois são parte integrante da prova.
5. No caderno de prova, o candidato poderá rabiscar, riscar, calcular, etc.
6. Os gabaritos preliminares da prova objetiva serão divulgados no dia 20/11/2012, até às 23h59min, nos sites www.fundatec.org.br e www.procergs.rs.gov.br.
7. Certifique-se de que este caderno contém 60 (sessenta) questões. Caso contrário, solicite ao fiscal da sala a sua substituição.

FUNDATEC .org.br

CONHECIMENTOS ESPECÍFICOS

QUESTÃO 01 - Associe as funções para *arrays* apresentadas na Coluna 1 a sua respectiva funcionalidade apresentada na Coluna 2.

Coluna 1
1. array_shift
2. array_splice
3. array_pop
4. array_push

Coluna 2
( ) Remove uma parcela do *array* e substitui com outros elementos.
( ) Retira o primeiro elemento de um *array*.
( ) Adiciona um ou mais elementos no final de um *array*.
( ) Retira um elemento do final do *array*.

A ordem correta de preenchimento dos parênteses, de cima para baixo, é:

A) 1 – 2 – 4 – 3.
B) 2 – 1 – 3 – 4.
C) 2 – 1 – 4 – 3.
D) 2 – 4 – 1 – 3.
E) 3 – 1 – 4 – 2.

QUESTÃO 02 - Considere os operadores de comparação em PHP. Preencha os parênteses com V quando a comparação for avaliada como verdadeira (*true)* e, com F quando a comparação for avaliada como falsa (*false)*.

( ) 0 == "fundatec"
( ) "100" == "1e2"
( ) "1" === 1
( ) "1" !== "1"
( ) 0 == false

A ordem correta de preenchimento dos parênteses, de cima para baixo, é:

A) V – V – F – F – V.
B) V – V – F – V – F.
C) F – F – F – F – V.
D) F – F – V – F – V.
E) V – V – V – V – F.

QUESTÃO 03 - Associe as funções de manipulação de *strings* de PHP apresentadas na Coluna 1 à respectiva descrição apresentada na Coluna 2.

Coluna 1
1. strcmp
2. strlen
3. substr
4. trim
5. strip_tags

Coluna 2
( ) Comparação de *string*.
( ) Remove os marcadores HTML e PHP de uma *string*.
( ) Retira espaço no início e final de uma *string*.
( ) Retorna o tamanho de uma *string*.
( ) Retorna uma parte de uma *string*.

A ordem correta de preenchimento dos parênteses, de cima para baixo, é:

A) 1 – 4 – 5 – 2 – 3.
B) 1 – 5 – 4 – 2 – 3.
C) 3 – 2 – 4 – 5 – 1.
D) 1 – 4 – 5 – 3 – 2.
E) 3 – 5 – 4 – 2 – 1.

QUESTÃO 04 - Associe as funções de sistema de arquivo apresentadas na Coluna 1 a sua respectiva descrição apresentada na Coluna 2.

Coluna 1
1. file
2. file_get_contents
3. fputs
4. feof
5. filesize

Coluna 2
( ) Testa pelo fim-de-arquivo.
( ) Lê todo o conteúdo de um arquivo para uma *string*.
( ) Retorna o tamanho do arquivo.
( ) Escrita *binary-safe* em arquivos.
( ) Lê todo o arquivo para um *array*.

A ordem correta de preenchimento dos parênteses, de cima para baixo, é:

A) 3 – 5 – 1 – 4 – 2.
B) 3 – 2 – 5 – 4 – 1.
C) 3 – 5 – 2 – 4 – 1.
D) 4 – 1 – 5 – 3 – 2.
E) 4 – 2 – 5 – 3 – 1.

QUESTÃO 05 - Considere o seguinte *script* PHP.

```
<?php

$i = 2;
$j = ++$i;
$k = $i++;
$m = --$k;
$n = $j-- + ++$m;

?>
```

Qual o valor armazenado na variável $*n* após a execução desse trecho de código?

A) 3
B) 4
C) 5
D) 6
E) 7

QUESTÃO 06 - Considere que a variável $*v* armazena o valor inteiro 4. Associe as expressões apresentadas na Coluna 1 ao respectivo resultado listado na Coluna 2. Considere que as operações são executadas independentemente uma das outras.

Coluna 1
1. $v >> 1
2. $v & 0x01
3. $v % 6

Coluna 2
( ) 4
( ) 0
( ) 2

A ordem correta de preenchimento dos parênteses, de cima para baixo, é:

A) 1 – 2 – 3.
B) 3 – 1 – 2.
C) 2 – 1 – 3.
D) 2 – 3 – 1.
E) 3 – 2 – 1.

QUESTÃO 07 - Sobre alguns conceitos de orientação a objetos em PHP, analise as assertivas abaixo.

I. O modificador *final* indica que o método necessita ser sobrescrito nas subclasses.
II. Ao se chamar um método em uma subclasse, PHP chama na ordem inversa todos os métodos sobrescritos das superclasses.
III. A instanciação de objetos de classes abstratas não é permitido.

Quais estão corretas?

A) Apenas I.
B) Apenas II.
C) Apenas III.
D) Apenas I e III.
E) Apenas II e III.

QUESTÃO 08 - Analise o seguinte *script* PHP.

```
<?php

$vetor = array(5,4,3,2,1);

for ( $i = 1; $i < sizeof($vetor); $i++ ) {
        $vetor[$i-1] = $vetor[$i];
        $vetor[$i] = $vetor[$i-1];
} // end for

?>
```

Após a execução desse trecho, qual será o conteúdo da variável *$vetor*?

A) 4, 3, 2, 1, 0
B) 4, 3, 2, 1, 1
C) 4, 3, 2, 1, null
D) 5, 4, 3, 2, 1
E) 5, 4, 3, 2, 0

QUESTÃO 09 - Analise o seguinte *script* PHP.

```
<?php

abstract class Letra {
	public function imprime() { echo("Sou uma letra.\n"); }
}

abstract class Consoante extends Letra {
	public function imprime() { echo("Sou uma consoante.\n"); }
}

abstract class Vogal extends Letra {
	public function imprime() { echo("Sou uma vogal.\n"); }
}

class A extends Vogal {
	public function imprime() { echo("A\n"); }
}

class B extends Consoante {
}

$a = new A();
$b = new B();

$a->imprime();
$b->imprime();

?>
```

Qual será a saída desse *script*?

A)

```
Sou uma letra.
Sou uma vogal.
A
Sou uma letra.
Sou uma consoante.
```

B)

```
Sou uma letra.
Sou uma vogal.
A
```

C)

```
A
```

D)

```
A
Sou uma consoante.
```

E)

```
Sou uma vogal.
Sou uma consoante.
```

QUESTÃO 10 - Sobre conceitos de orientação a objetos em PHP, analise as assertivas abaixo.

I. Uma classe pode implementar múltiplas interfaces.
II. Apenas herança simples é suportada (i.e. uma classe pode estender apenas uma classe base).
III. Uma classe declarada com o modificador final não pode ser estendida.

Quais estão corretas?

A) Apenas II.
B) Apenas I e II.
C) Apenas I e III.
D) Apenas II e III.
E) I, II e III.

QUESTÃO 11 - Selecione a alternativa que apresenta uma função PHP que calcula e retorna o somatório de todos os elementos com índice ímpar de um vetor (*array*) de inteiros.

A)

```
function fa($v) {
	$resultado = 0;
	for ( $i = 0; $i < sizeof($v); $i++ ) {
		$resultado += $v[$i];
	}
	return $resultado;
}
```

B)

```
function fb($v) {
	$resultado = 0;
	for ( $i = 0; $i < sizeof($v); $i++ ) {
		if ( $i % 2 == 1)
			$resultado += $v[$i];
	}
	return $resultado;
}
```

C)

```
function fc($v) {
	$resultado = 0;
	foreach ( $v as $i ) {
		$resultado += $i;
	}
	return $resultado;
}
```

D)

```
function fd($v) {
	$resultado = 0;
	$flag = true;
	foreach ( $v as $i ) {
		if ( $flag )
			$resultado += $i;
		$flag = !$flag;
	}
	return $resultado;
}
```

E)

```
function fe($v) {
	$resultado = 0;
	$i = 0;
	while ( ++$i < sizeof($v) ) {
		if ( $i % 2 == 0 )
			$resultado += $v[$i];
	}
	return $resultado;
}
```

QUESTÃO 12 - Selecione a alternativa que melhor e mais amplamente descreve a seguinte construção PHP.

```
list( $v1, $v2, $v3, , $v5) = array( 1, 2, 3, 4, 5);
```

A) A criação de um *array* com os valores contidos nas variáveis *$v1*, *$v2*, *$v3* e *$v5*.
B) Comparação entre uma lista e um *array*.
C) A criação de uma lista com valores 1, 2, 3, 4, NULL e 5.
D) Atribuição dos valores 1, 2, 3 e 5, respectivamente, às variáveis *$v1, $v2, $v3 e $v5.*
E) A criação de uma lista com valores 1, 2, 3, 4 e 5.

QUESTÃO 13 - Uma maneira correta de representar a equação algébrica abaixo em código PHP é dada pela alternativa:

$$\frac{-b + \sqrt{b^2 - 4ac}}{2a}$$

A) (-$b + sqrt(pow($b,2) - 4*$a*$c)) / 2*$a
B) -$b + sqrt($b*$b - 4*$a*$c) / (2*$a)
C) (-$b + sqrt($b*$b - 4*$a*$c)) / (2*$a)
D) (-$b + sqrt($b*$b - 4*$a*$c)) / 2*$a
E) -($b + sqrt(pow($b,2) - 4*$a*$c)) / 2*$a

QUESTÃO 14 - Considere o seguinte *script* PHP.

```
<?php

class MyClass {
	private $a = 0;
	private $b = 1;
	private $c = 2;
	private $d = 3;
	private $e = 4;

	public function getA() { return $this->a; }
	public function getB() { return $this->b; }
	public function getC() { return $this->c; }
	public function getD() { return $this->d; }
	public function getE() { return $this->e; }

	public function setA($value) { $this->a = $value; }
	public function setB($value) { $this->b = $value; }
	public function setC($value) { $this->c = $value; }
	public function setD($value) { $this->d = $value; }
	public function setE($value) { $this->e = $value; }

}

$stuff = range('A', 'E');
$str0 = "set$stuff[3]";
$str1 = "get$stuff[2]";
$str2 = "get$stuff[3]";

$obj = new MyClass();
$obj->$str0( $obj->$str1() + 10 );
$v = $obj->$str2();

?>
```

Ao final da execução desse *script*, qual valor conterá a variável *$v*?

A) 10
B) 11
C) 12
D) 13
E) 14

QUESTÃO 15 - Analise as assertivas abaixo sobre exceções em PHP.

I. Se uma exceção não for capturada por um bloco *try/catch*, PHP lançará um Erro Fatal a menos que um tratador seja definido através da função *set_exception_handler*.
II. Após a execução do tratador de exceção definido por *set_exception_handler* ser chamado devido ao lançamento de uma exceção, PHP termina a execução do *script*.
III. Exceções podem ser lançadas dentro de um bloco *catch*.

Quais estão corretas?

A) Apenas I.
B) Apenas II.
C) Apenas III.
D) Apenas I e III.
E) Apenas II e III.

QUESTÃO 16 - Sobre passagem de parâmetros para funções em PHP, analise as assertivas abaixo.

I. PHP possui somente o mecanismo de passagem de parâmetro por valor.
II. Chamar uma função utilizando-se mais parâmetros do que os definidos explicitamente pela função gera um Erro Fatal.
III. Os parâmetros em PHP podem possuir um valor padrão.

Quais estão corretas?

A) Apenas I.
B) Apenas III.
C) Apenas I e III.
D) Apenas II e III.
E) I, II e III.

QUESTÃO 17 - Sobre visibilidade de métodos e propriedades de classes em PHP, analise as assertivas abaixo.

I. Métodos e propriedades privados (definidos com modificador *private*) não podem ser acessados diretamente nas subclasses.
II. Uma propriedade definida com o modificador *portected* só pode ser acessada por classes definidas no mesmo arquivo.
III. Se nenhum modificador de visibilidade (*private*, *public* ou *protected*) for explicitado para um método, tal método será considerado público.

Quais estão corretas?

A) Apenas I.
B) Apenas II.
C) Apenas III.
D) Apenas I e II.
E) Apenas I e III.

QUESTÃO 18 - Sobre *arrays* na linguagem PHP, analise as assertivas a seguir.

I. A sintaxe *$v[ ] = 10* é válida e indica que o valor *10* será inserido no *array $v*.
II. Para remover um elemento de um *array*, pode ser usado o comando *unset*.
III. Caso a chave *nome* já exista num vetor *$v*, o comando *$v["nome"] = "Fundatec"* gera uma exceção.

Quais estão corretos?

A) Apenas I.
B) Apenas II.
C) Apenas III.
D) Apenas I e II.
E) Apenas II e III.

QUESTÃO 19 - Considere algumas funções matemáticas de PHP. Qual o valor mais aproximado da expressão na caixa abaixo?

```
sqrt( -ceil( max(-16.1, -4.2 ) ) + 0.5 )
```

A) 5.5166334660037
B) 4.1833001326704
C) 4.062019202318
D) 3.0822070014845
E) 2.1213203435596

QUESTÃO 20 - Suponha que $*i*, $*j*, $*k* e $*m* sejam variáveis do tipo inteiro e que *$i = 1*, *$j = 6*, *$k = 4* e *$m = -7*. Preencha os parênteses com V, quando a expressão for avaliada como verdadeira (*true),* ou com F, quando a expressão for avaliada como falsa (*false)*.

( ) $m > -3 && $i >= 5
( ) 2*2 == $k || $j != 5
( ) $m < -8 || ($k == -$m + 1)
( ) $k >= $j? false : true

A ordem correta de preenchimento dos parênteses, de cima para baixo, é:

A) F – F – V – V.
B) F – V – V – F.
C) V – F – F – V.
D) F – V – F – V.
E) V – F – V – F.

QUESTÃO 21 - Considere o seguinte *script*.

```
<?php

define( 'N', 5 );

for ( $i = 0; $i < N; $i++ ) {
        for ( $j = 0; $j < ($i < N/2? $i+1 : N-$i); $j++ ) {
                echo "*";
        } // end for
        echo "\n";
} // end for

?>
```

A saída gerada por esse *script* é correspondente à alternativa

A)
*****
****
***
**
*

B)
*
**
***
****
*****

C)
*
**
***
**
*

D)
*****
*****
*****
*****
*****

E)
***
**
*
**
***

QUESTÃO 22 - Considere o *script* PHP apresentado na caixa abaixo.

```
<?php

class A {
        public $value = 0;
        public function __constructor() {
                $value = 1;
        }
}

class B {
        public $value = 0;
        public function __constructor() {
                $value = 2;
        }
}

$a = new A();
$b = new B();

$a->value++;
$b->value--;

$result = $a->value + $b->value;

?>
```

Qual o valor armazenado na variável *$result* ao final da execução do *script*?

A) 0
B) 1
C) 2
D) 3
E) 4

QUESTÃO 23 - Qual função PHP calcula corretamente o fatorial de um número inteiro positivo?

A)

```
function fa($n) {
        return $n!;
}
```

B)

```
function fb($n) {
        return $n * fb($n-1);
}
```

C)

```
function fc($n) {
        $result = 0;
        while($n > 0)
                $result *= $n--;
        return $result;
}
```

D)

```
function fd($n) {
        return $n >=0 ? $n * fd($n-1) : 1;
}
```

E)

```
function fe($n) {
        $result = 1;
        for ( $i = 1; $i <= $n; $i++)
                $result *= $i;
        return $result;
}
```

QUESTÃO 24 - Qual será a saída do seguinte *script* PHP?

```
<?php

$a = 15;

switch ($a) {
        case 0x15:
                echo("A"); break;
        case 0xF:
                break;
        default:
                echo("B");
}

if ( $a <= 0 )
        echo ("C");
        if ($a > 15)
                echo("D");
else
        echo("E");

?>
```

A) A
B) B
C) C
D) D
E) E

QUESTÃO 25 - Analise o seguinte *script* PHP:

```
<?php

class MyCoolException extends Exception {
}

try {
        throw new MyCoolException();
        echo("0");
} catch (Exception $e) {
        echo("1");
} catch (MyCoolException $e) {
        echo("2");
}

?>
```

Qual será a saída desse *script*?

A) 1
B) 2
C) 01
D) 12
E) 012

QUESTÃO 26 - Sobre o método construtor de classes em PHP, assinale a alternativa correta.

A) Precisam ser definidos explicitamente para todas as classes.
B) Os construtores das superclasses não são chamados implicitamente pelo construtor da subclasse.
C) Pode ser sobrecarregado, redeclarando-o com diferente número de parâmetros.
D) Não pode receber parâmetros.
E) O método construtor é chamado assim que todas as referências a um objeto particular são removidas.

QUESTÃO 27 - Sobre o método destrutor de classes em PHP, assinale a alternativa correta.

A) Precisam ser definidos explicitamente para todas as classes.
B) Os destrutores das superclasses são chamados implicitamente pelo destrutor da subclasse.
C) Pode ser sobrecarregado, redeclarando-o com diferente número de parâmetros.
D) O nome do destrutor precisa iniciar com ~ (til) seguido do nome da classe.
E) O nome do método destrutor deve ser __destruct.

QUESTÃO 28 - Sobre a palavra-chave *parent* em PHP, analise as assertivas a seguir:

I. Pode ser usada numa subclasse para chamar o construtor da superclasse imediata.
II. Pode ser usada, seguida do operador ->, para acessar membros da superclasse imediata.
III. Pode ser utilizada para acessar membros *protected* da superclasse imediata.

A) Apenas II.
B) Apenas III.
C) Apenas I e III.
D) Apenas II e III.
E) I, II e III.

QUESTÃO 29 - Sobre interfaces em PHP, analise as seguintes assertivas.

I. Nas interfaces, os métodos são apenas especificados, mas não implementados.
II. Todos os métodos declarados em uma interface devem ser públicos.
III. Uma classe pode implementar mais de uma interface.

Quais estão corretas?

A) Apenas I.
B) Apenas II.
C) Apenas I e II.
D) Apenas II e III.
E) I, II e III.

QUESTÃO 30 - Sobre sobrecarga de métodos e operadores em PHP, analise as assertivas a seguir:

I. É possível sobrecarregar um método redeclarando-o na mesma classe com diferentes parâmetros.
II. O método *__call()* é disparado quando se invoca métodos inacessíveis em um contexto de objeto.
III. PHP possui o mecanismo de sobrecarga de operadores.

Quais estão corretas?

A) Apenas I.
B) Apenas II.
C) Apenas III.
D) Apenas I e II.
E) Apenas II e III.

QUESTÃO 31 - Considere algumas estruturas de dados apresentadas na Coluna 1 e relacione-as com a respectiva descrição na Coluna 2.

Coluna 1
1. Fila
2. Pilha
3. Árvore
4. Tabela *Hash*

Coluna 2
( ) Associa chaves de pesquisa a valores.
( ) Grafo acíclico.
( ) FIFO (primeiro a entrar, primeiro a sair).
( ) LIFO (último a entrar, primeiro a sair).

A ordem correta de preenchimento dos parênteses, de cima para baixo, é:

A) 2 – 4 – 1 – 3.
B) 3 – 4 – 2 – 1.
C) 3 – 2 – 1 – 4.
D) 4 – 3 – 2 – 1.
E) 4 – 3 – 1 – 2.

QUESTÃO 32 - Associe as funções para *arrays* apresentadas na Coluna 1 a sua respectiva funcionalidade na Coluna 2.

Coluna 1
1. reset
2. each
3. key
4. current
5. prev

Coluna 2

( ) Retorna o par chave/valor corrente de um *array* e avança o seu cursor.
( ) Retrocede o ponteiro interno de um *array*.
( ) Retorna a chave da posição atual de um *array*.
( ) Faz o ponteiro interno de um *array* apontar para o seu primeiro elemento.
( ) Retorna o elemento apontado pelo ponteiro interno de um *array*.

A ordem correta de preenchimento dos parênteses, de cima para baixo, é:

A) 2 – 4 – 3 – 1 – 5.
B) 5 – 2 – 1 – 3 – 4.
C) 2 – 5 – 3 – 1 – 4.
D) 3 – 4 – 2 – 1 – 5.
E) 2 – 5 – 1 – 3 – 4.

QUESTÃO 33 - Considere a precedência de operadores em PHP. Preencha os parênteses com números de 1 a 5, em que 1 indica o operador de maior precedência (que é executado primeiro) e 5 o operador de menor precedência.

( ) * (multiplicação)
( ) -- (pós-decrementado)
( ) + (soma)
( ) && (e condicional)
( ) || (ou condicional)

A ordem correta de preenchimento dos parênteses, de cima para baixo, é:

A) 2 – 1 – 3 – 4 – 5.
B) 2 – 1 – 3 – 5 – 4.
C) 1 – 2 – 3 – 4 – 5.
D) 1 – 3 – 2 – 5 – 4.
E) 1 – 3 – 2 – 4 – 5.

QUESTÃO 34 - Considere o seguinte *script* PHP.

```
<?php

$arrayv = array(2, 3, 5, 7, 11, 13, 17);

$arrayi = array(1, 2, 3, 0);
$arrayj = array(1, 4, 4, 5);

$s = 0;
for ( $i = 0; $i < count($arrayi); $i++ )
        $s += $arrayv[$arrayi[$i]] + $arrayv[$arrayj[$i]];

echo($s);

?>
```

Qual será a saída desse *script*?

A) 51
B) 55
C) 57
D) 64
E) 70

QUESTÃO 35 - Sobre expressões regulares, analise o seguinte *script* PHP.

```
<?php

$pattern = "/a[bc]+/";

$i = 0;
$i += preg_match($pattern, "aaaaab")? 1 : 0;
$i += preg_match($pattern, "a")? 1 : 0;
$i += preg_match($pattern, "bcbc")? 1 : 0;
$i += preg_match($pattern, "acccc")? 1 : 0;
$i += preg_match($pattern, "cbbbc")? 1 : 0;

echo($i);

?>
```

Qual será a saída desse *script*?

A) 1
B) 2
C) 3
D) 4
E) 5

QUESTÃO 36 - Sobre expressões regulares em PHP, considere que se deseja casar a sequência de caracteres apresentada na caixa a seguir.

```
<a href='mailto:abc@fundatec.org.br'>Envie um E-mail</a>
```

Qual padrão (expressão regular) poderá ser usado para identificar a sequência através da função *preg_match*?

A) /<a\s*href\s+=\s+'mailto:.*'\s+>.*<\/a>/
B) /<a\s+href\s*=\s*'*'\s*>[a-zA-Z0-9@]*<\/a>/
C) /<a\s+href\s+=\s+'mailto:*'\s+>*<\/a>/
D) /<a\s+href\s*=\s*'.*'\s*>.*<\/a>/
E) /<a\s*href\s*=\s*'[a-zA-Z0-9@]*'\s*>.*<\/a>/

QUESTÃO 37 - Considere a função recursiva *f* implementada em PHP apresentada na caixa abaixo.

```
function f($n) {
        if ( $n > 1 )
                return f($n-1) + f($n-2);
        else
                return $n;
}
```

Qual o valor de *f(4)*?

A) 2
B) 3
C) 4
D) 5
E) 8

QUESTÃO 38 - Considere o *script* PHP na caixa abaixo.

```
<?php

$n = 1;

function f() {$n = 0;}

$n++;
f();
$n += 3;
f();
$n--;

?>
```

Qual o valor da variável *$n* ao final da execução desse script?

A) 1
B) -1
C) 0
D) 3
E) 4

QUESTÃO 39 - Considere a função recursiva implementada em PHP apresentada na caixa abaixo.

```
function f($n) {
        if ( $n == 0 )
                return 0;
        else
                return $n + f($n-1);
}
```

Supondo *$n > 0*, tal função pode ser traduzida para a função *não recursiva* apresentada em qual alternativa?

A)

```
function fa($n) {
        return $n==0 ? 0 : $n + fa($n-1);
}
```

B)

```
function fb($n) {
        $r = 0;
        while(--$n > 0)
                $r += $n+1;
        return $r;
}
```

C)

```
function fc($n) {
        $r = 0;
        for ( $i = 0; $i <= $n; $i++)
                $r += $i;
        return $r;
}
```

D)

```
function fd($n) {
        if ( $n > 0 )
                return $n + fd($n-1);
        else
                return 0;
}
```

E)

```
function fe($n) {
        return (2*($n-1))/2;
}
```

QUESTÃO 40 - Suponha que se queira tratar a *string* de consulta (em inglês, *query string*) do seguinte tipo de URL:

```
http://www.fundatec.org.br/40.php?id=1
```

Personalizando o conteúdo da página na caixa abaixo, de acordo com o parâmetro *id*.

```
<html>
<head>
<title>Questão 40</title>
</head>

<body>
Bem-vindo candidato _
</body>

</html>
```

Para que seja exiba a mensagem “Bem-vindo candidato 1” no corpo da página, qual trecho de código PHP pode substituir o caractere _ (sublinhado)?

A) $id
B) <?= $_GET['id'] ?>
C) <?= $_POST['id'] ?>
D) <?php $_GET['id'] ?>
E) $_POST['id']

QUESTÃO 41 - Considere a seguinte função JavaScript:

```
function f(a) {
        return function(b) {
                return a * b;
        };
}
```

Qual o valor da expressão *f((f(3))(4))(2)*?

A) 6
B) 8
C) 9
D) 12
E) 24

QUESTÃO 42 - Considere que o arquivo texto *input.txt* tenha o seguinte conteúdo:

```
Lista de Times Brasileiros Campeões da Libertadores
Santos 3 0
São Paulo 3 2
Cruzeiro 2 1
Grêmio 2 1
Internacional 2 1
Vasco 1 0
Flamengo 1 3
Palmeiras 1 1
Corinthians 1 0
```

E que o seguinte script PHP é utilizado para lê-lo (suponha que o arquivo esteja no mesmo diretório e acessível ao *script*).

```
<?php

$lines = file('input.txt');

$sum = 0;
for ( $i = 1; $i < count($lines); $i++ ) {
        if ( preg_match( "/.*\s+([0-9]+)\s+([01])/", $lines[$i], $matches ) ) {
                if ( $matches[2] > 0 )
                        $sum += $matches[1];
        }
}

echo $sum;

?>
```

Qual será a saída desse *script*?

A) 16
B) 5
C) 11
D) 7
E) 9

QUESTÃO 43 - Sobre *cookies* e gerenciamento de *cookies* em PHP, considere as assertivas abaixo:

I. *Cookies* são um mecanismo para guardar dados no navegador remoto.
II. A função *setcookie* precisa ser chamada antes que qualquer outro dado seja enviado ao navegador para que os *cookies* sejam definidos corretamente.
III. Os *cookies* são enviados juntamente com o corpo da mensagem HTTP.

Quais estão corretas?

A) Apenas I.
B) Apenas II.
C) Apenas I e II.
D) Apenas I e III.
E) Apenas II e III.

QUESTÃO 44 - Para que um servidor PHP sirva corretamente um arquivo do tipo JSON, qual comando deve ser incluído no início do *script*?

A) setcookie('Content-type', 'application/json')
B) header('Content-type: application/json')
C) echo 'Content-type: application/json'
D) exit 'Content-type: application/json'
E) http_head('Content-type: application/json')

QUESTÃO 45 - Considere a seguinte página HTML.

```
<html>
<head>
	<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />

	<title>Questão 45</title>

	<script>
		function load() {
			var e = document.getElementById('mydiv');

			(function f1() { e.innerHTML = "Oi, Terra!"; });
			(function f2() { e.innerHTML = "Oi, Lua!"; });
			(function f3() { e.innerHTML = "Oi, Sol!"; })();
			(function f4() { e.innerHTML = "Oi, Marte!"; });
		}
	</script>

<head>

<body onload='load()'>
	<div id='mydiv'>Oi, Mundo!</div>
</body>

</html>
```

Após o carregamento completo da página, qual a mensagem exibida no elemento com *id* igual a *mydiv*?

A) Oi, Terra!
B) Oi, Lua!
C) Oi, Sol!
D) Oi, Marte!
E) Oi, Mundo!

QUESTÃO 46 - Sobre sessões em PHP, considere as assertivas a seguir.

I. Os dados da sessão são armazenados no navegador.
II. A função *session_start* precisa ser chamada antes que qualquer outro dado seja enviado ao navegador quando o PHP está configurado para usar *cookies* para gerenciar as sessões.
III. A variável superglobal $_REQUEST é utilizada para armazenar os dados da sessão.

Quais estão corretas?

A) Apenas I.
B) Apenas II.
C) Apenas III.
D) Apenas I e II.
E) Apenas II e III.

QUESTÃO 47 - Sobre chaves-primárias e chaves-estrangeiras em banco de dados relacional SQL, analise as seguintes assertivas.

I. Uma chave-primária pode ser composta de mais de um campo.
II. Registros diferentes, numa mesma tabela, podem ter o mesmo valor de chave-primária.
III. Uma chave-estrangeira entre duas tabelas pode relacionar apenas as chaves-primárias das duas tabelas.

Quais estão corretas?

A) Apenas I.
B) Apenas II.
C) Apenas III.
D) Apenas I e II.
E) I, II e III.

QUESTÃO 48 - Considere o seguinte *script* JavaScript.

```
var name;
var age;

function Person(myName, myAge) {
        this.getName = function() { return name; }
        this.getAge = function() { return age; }

        this.setName = function(value) { name = value; }
        this.setAge = function(value) { age = value; }

        name = myName;
        age = myAge;

}

var person0 = new Person("João", 54);
var person1 = new Person("Maria", 62)

person0.setName( person0.getName() + ' Félix');

alert(name);
alert(person0.getName());
alert(person1.getAge());
```

Qual será o conteúdo respectivamente das três mensagens de alerta?

A) *null*, *João* e *62*
B) *undefined*, *João* e *54*
C) null, João Félix, 62
D) *Maria Félix, Maria Félix e 62*
E) *undefined*, *João Félix* e *62*

Para responder às questões 49 a 55, considere as figuras abaixo.

id	nome	grupo	salario	sexo
1	João	NULL	2689.00	1
2	Maria	1	1555.00	0
3	José	2	NULL	1
4	Paulo	1	2700.00	1
5	Ana	2	3214.00	0
6	Cris	NULL	5700.00	0
7	Marco	1	4000.00	1
8	Fernanda	2	1256.00	0
9	Pedro	2	NULL	1
10	Madalena	1	8500.00	0

Figura 1 - Tabela de dados *pessoal*.

id	nome
1	Grupo 1
2	Grupo 2
3	Grupo 3

Figura 2 - Tabela de dados *grupos*.

QUESTÃO 49 - Considerando a tabela *pessoal* mostrada na Figura 1, quantos registros retornam o comando SQL na caixa abaixo?

```
SELECT grupo, AVG(salario) FROM pessoal GROUP BY grupo HAVING grupo IS NOT NULL;
```

A) Nenhum
B) 1
C) 2
D) 3
E) 10

QUESTÃO 50 - Considerando a tabela *pessoal* mostrada na Figura 1, qual o valor retornado pela consulta SQL na caixa abaixo?

```
SELECT COUNT(*) FROM pessoal WHERE sexo = 0 AND salario >= 2000;
```

A) 0
B) 1
C) 2
D) 3
E) 4

QUESTÃO 51 – Considerando a tabela *pessoal* mostrada na Figura 1, quantos registros serão retornados pela consulta SQL abaixo?

```
SELECT * FROM
        pessoal AS t0, pessoal AS t1
WHERE
        t0.sexo = 0 AND t0.sexo != t1.sexo;
```

A) 5
B) 10
C) 25
D) 50
E) 100

QUESTÃO 52 - Considerando a tabela *pessoal* mostrada na Figura 1 e que o campo id é chave-primária dessa tabela, analise o comando SQL na caixa abaixo e marque a alternativa correta.

```
INSERT INTO pessoal
(id, nome, grupo, salario, sexo)
VALUES
(11, "Madalena", 1, 1888, 0);
```

A) Após a execução, a tabela *pessoal* conterá 11 registros.
B) O registro com o campo *nome* igual a Madalena será atualizado.
C) Contém erro de sintaxe.
D) Após a execução, a tabela *pessoal* conterá 9 registros.
E) O novo registro não será inserido devido à chave-primária duplicada.

QUESTÃO 53 - Considerando a tabela *pessoal* mostrada na Figura 1 e a tabela *grupos* mostrada na Figura 2, quantos registros serão retornados pela consulta SQL na caixa abaixo?

```
SELECT * FROM grupos INNER JOIN pessoal ON pessoal.grupo = grupos.id;
```

A) 30
B) 20
C) 10
D) 9
E) 8

QUESTÃO 54 - Considere a tabela *pessoal* mostrada na Figura 1. Deseja-se utilizar uma consulta SQL para atualizar todos e somente os registros cujo campo grupo *não* é nulo (NULL), alterando o valor do campo grupo para 4. Para tanto, é possível utilizar qual comando SQL?

A) UPDATE pessoal SET grupo = 4;
B) UPDATE pessoal SET grupo = 4 WHERE grupo = 0;
C) UPDATE pessoal SET grupo = 4 WHERE grupo IS NULL;
D) UPDATE pessoal SET grupo = 4 WHERE grupo IS NOT NULL;
E) UPDATE pessoal SET grupo = 4 WHERE grupo <> NULL;

QUESTÃO 55 - Considere a tabela *pessoal* mostrada na Figura 1 e a tabela *grupos* mostrada na Figura 2. Quais registros, referenciados pelo valor do campo *id*, são retornados pela consulta SQL abaixo, caso se considere a ordem?

```
SELECT * FROM
        pessoal, grupos
WHERE
        pessoal.grupo = grupos.id AND
        grupos.id = 2
ORDER BY
        pessoal.salario DESC;
```

A) 5, 8, 3, 9
B) 9, 3, 8, 5
C) 8, 5
D) 5, 8
E) A consulta retorna nenhum (zero) registro.

QUESTÃO 56 - Suponha que um site use a seguinte consulta SQL para excluir um registro da tabela *users*:

```
DELETE FROM users WHERE id = $id
```

Se o conteúdo de *$id* não for tratado corretamente, um atacante pode fazer uso da técnica de injeção SQL (em inglês, *SQL injection*) para alterar o comportamento esperado da consulta. Qual alternativa apresenta uma *string* potencialmente perigosa que, quando substitui a variável *$id*, altera o comportamento da consulta acima para excluir todos os registros da tabela?

A) *
B) TRUE
C) 0 OR TRUE
D) -- OR TRUE
E) IS NOT NULL

QUESTÃO 57 - Considere as assertivas abaixo sobre o sistema de banco de dados MySQL.

I. O tipo padrão de tabela MyISAM fornece suporte a transações.
II. O tipo de tabela InnoDB provê suporte a chaves-estrangeiras.
III. Uma única instrução *update* é atômica, i.e. não pode ser interrompida por outra consulta ou executada pela metade.

Quais estão corretas?

A) Apenas II.
B) Apenas III.
C) Apenas I e III.
D) Apenas II e III.
E) I, II e III.

QUESTÃO 58 - Considere que a seguinte página HTML é executada em um navegador moderno com suporte ao objeto XMLHttpRequest.

```
<!DOCTYPE html>
<html>
<head>
	<script>
		function ajax() {
			var xmlhttp;
			if (window.XMLHttpRequest) {
				xmlhttp= new XMLHttpRequest();
			}

			if (xmlhttp) {
				var params = "op1=Fundatec&op0=Concursos";

				xmlhttp.onreadystatechange = function() {
					var success =
						xmlhttp.readyState == 4 &&
						xmlhttp.status == 200;

					if ( success ) {
						var e = document.getElementById("content");
						e.innerHTML = xmlhttp.responseText;
					}
				}
				xmlhttp.open("POST","58.php",true);
				xmlhttp.setRequestHeader("Content-type",
					"application/x-www-form-urlencoded");
				xmlhttp.send(params);
			} else {
				alert("Navegador não suporta AJAX.");
			}
		}
	</script>

</head>
<body>

	<div id="content"></div>
	<button type="button" onclick="ajax()">Enviar</button>

</body>
</html>
```

Considere também que, no mesmo domínio, encontra-se o script PHP *58.php* mostrado na caixa abaixo.

```
<?php

$op0 = @$_POST['op0'];
$op1 = @$_POST['op1'];

$value = strcmp($op0, $op1);

if ( $value < 0 )
	die("$op0 $op1");
else if ( $value > 0 )
	die("$op1 $op0");
else
	die("$op0");

echo " OK";

?>
```

Suponha que não ocorram erros na requisição e que o servidor responda corretamente com status HTTP OK. Após se clicar no botão Enviar e a requisição AJAX ter sido completada, qual o conteúdo do elemento *div* com *id* igual a *content*?

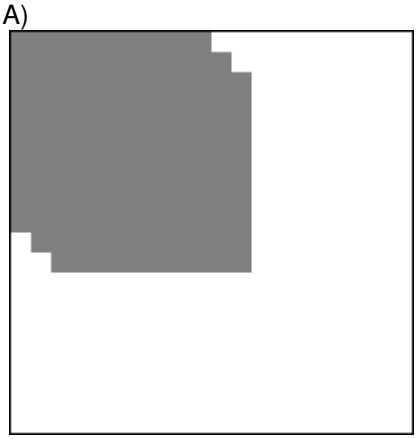
A) Fundatec Concursos
B) Concursos Fundatec
C) Fundatec Concursos OK
D) Concursos Fundatec OK
E) OK

QUESTÃO 59 - Considere a seguinte página HTML:

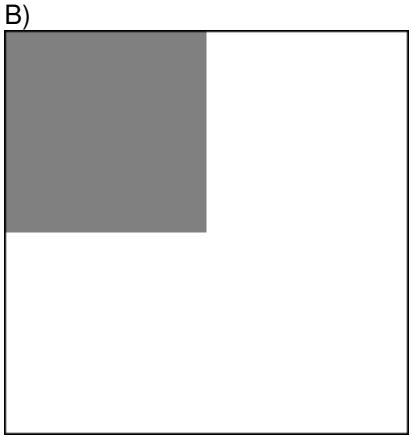
```
<html>
<head>
	<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />

	<title>Questão 59</title>

	<style>
		.box {
			top:10px;
			left:10px;
			width:100px;
			height:100px;
			background-color: gray;
		}

		#container {
			width: 200px;
			height: 200px;
			border: 1px solid black;
		}

		#div1 { position: relative; }
		#div3 { position: relative; }
	</style>

<head>

<body>
	<div id='container'>
		<div id='div0' class='box'>
			<div id='div1' class='box'>
				<div id='div2' class='box'>
					<div id='div3' class='box'>
						<div id='div4' class='box'>
						</div>
					</div>
				</div>
			</div>
		</div>
	</div>
</body>

</html>
```

Qual alternativa melhor representa o leiaute dessa página?

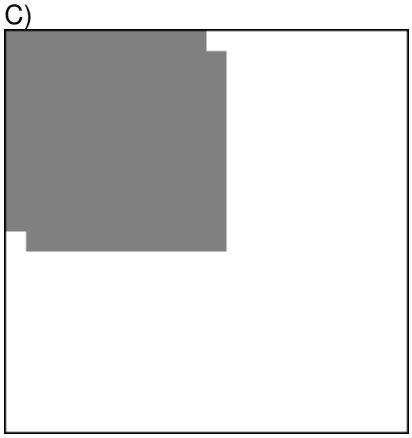
A)



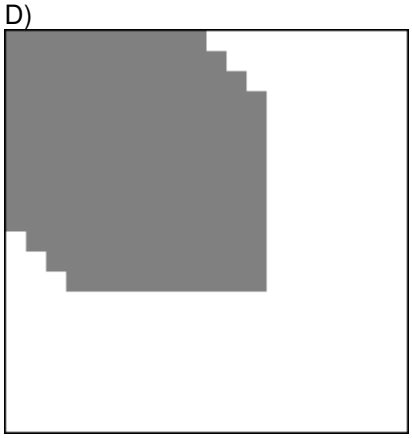
B)



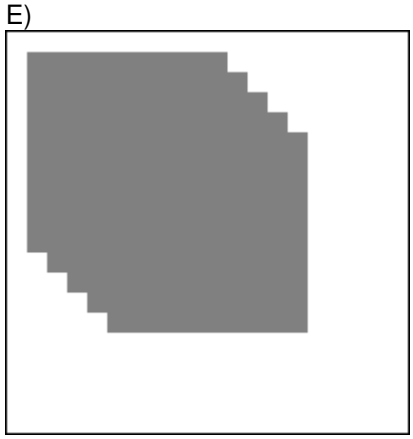
C)



D)



E)



QUESTÃO 60 – Sobre o comando *clone* de PHP, analise as assertivas abaixo.

I. No comportamento padrão, o comando *clone* replica superficialmente todas as propriedades de um objeto. Propriedades que são referencias a objetos continuam referenciando os mesmos objetos.
II. Pode ter seu comportamento alterado implementando-se o método *__clone*.
III. Ao clonar um objeto, o método construtor é chamado para o novo objeto.

Quais estão corretas?

A) Apenas I.
B) Apenas II.
C) Apenas I e II.
D) Apenas I e III.
E) I, II e III.