


CIDADES DE CHARQUEADAS, GRAVATAÍ, JAGUARÃO E SANTANA DO
LIVRAMENTO

INSTRUÇÕES GERAIS

- 1 - Este caderno de prova é constituído por 40 (quarenta) questões objetivas.
- 2 - A prova terá duração máxima de 04 (quatro) horas.
- 3 - Para cada questão, são apresentadas 04 (quatro) alternativas (a – b – c – d).
APENAS UMA delas responde de maneira correta ao enunciado.
- 4 - Após conferir os dados, contidos no campo Identificação do Candidato no Cartão de Resposta, assine no espaço indicado.
- 5 - Marque, com caneta esferográfica azul ou preta de ponta grossa, conforme exemplo abaixo, no Cartão de Resposta – único documento válido para correção eletrônica.


- 6 - Em hipótese alguma, haverá substituição do Cartão de Resposta.
- 7 - Não deixe nenhuma questão sem resposta.
- 8 - O preenchimento do Cartão de Resposta deverá ser feito dentro do tempo previsto para esta prova, ou seja, 04 (quatro) horas.
- 9 - Serão anuladas as questões que tiverem mais de uma alternativa marcada, emendas e/ou rasuras.
- 10 - O candidato só poderá retirar-se da sala de prova após transcorrida 01 (uma) hora do seu início.

BOA PROVA!

1. Analise o trecho de código abaixo.

```
if (saudacao<=200)
    if (saudacao<100)
        if (saudacao<0) printf("Oi!");
        else printf("Olá!");
    else printf("Tudo bem?");
else printf("Como vai você?");
```

Qual faixa de valores contidos na variável do tipo inteiro saudacao exibe a expressão "Tudo bem?" na tela?

- a) $saudacao \leq 100$
- b) $0 < saudacao < 100$
- c) $100 < saudacao \leq 200$
- d) $100 \leq saudacao \leq 200$

2. Considere a função abaixo para num=10.

Quantas chamadas recursivas ocorrem, desconsiderando a primeira chamada da função?

```
int fat(int num) {
    if (num==1) return num;
    else return(num * fat(num-1));
}
```

- a) 8
- b) 9
- c) 10
- d) 11

3. Observe o procedimento abaixo, escrito em C.

```
void procedimento(int a, int b) {
    a = a + b;
    b = a - b;
    a = a - b;
}
```

O que esse procedimento faz?

- a) Inverte os valores de a e b.
- b) Não altera os valores de a e b.
- c) Calcula o máximo divisor comum entre a e b.
- d) Divide b por a utilizando a soma e as subtrações efetuadas.

4. Observe o seguinte programa escrito em C.

```
#include <stdio.h>
int main(void) {
    int vetor[] = {20,30,40};
    int *p;
    p = &vetor[1];
    p++;
    printf("%d - ",*p);
    (*p)++;
    printf("%d - ",*p);
    printf("%d - %d - %d",vetor[0],vetor[1],vetor[2]);
    return 0;
}
```

Qual a saída gerada em tela por esse programa?

- a) 30 - 31 - 20 - 30 - 41
- b) 40 - 41 - 20 - 30 - 41
- c) 30 - 31 - 20 - 31 - 40
- d) 40 - 41 - 20 - 31 - 40

5. Observe o seguinte programa escrito em C.

```
#include <stdio.h>
int num;
int funcao(int num, int x) {
    int a=0, b=1, c;
    if (num < 2) {
        c = num;
        x = 0;
    } else {
        c = funcao(num-1,a) + funcao(num-2,b);
        x = a + b + 1;
    }
    printf("%d %d; ",num,x);
    return c;
}
int main(void) {
    num = 3;
    printf("%d %d",funcao(3,num),num);
    return 0;
}
```

Após a execução desse programa quais valores serão impressos em tela?

- a) 1 3; 0 1; 2 0; 1 4; 3 2; 0 3
- b) 1 0; 0 0; 2 2; 0 0; 2 2; 1 3
- c) 1 0; 0 0; 2 2; 1 0; 3 2; 2 3
- d) 0 3; 0 0; 2 2; 1 0; 3 3; 2 0

6. Considere o texto a seguir.

A acessibilidade ou visibilidade é um aspecto de extrema importância na programação orientada a objetos. O especificador _____ indica que o campo ou método só pode ser usado na implementação de subclasse. Já o especificador _____ indica que o campo ou método pode ser utilizado livremente, e o especificador _____ indica que campos e métodos devem ficar ocultos, ou seja, não podem ser utilizados fora da implementação da própria classe.

As palavras que completam corretamente as lacunas são, respectivamente,

- a) public, protected e private
- b) private, public e protected
- c) protected, public e private
- d) protected, private e public

7. Na programação orientada a objetos, a herança permite uma forma de reutilização de software em que uma nova classe pode herdar membros de uma classe existente. Sobre herança, são feitas as seguintes afirmações:

- I. O tempo de desenvolvimento do programa pode se reduzido.
- II. Cada objeto de uma subclasse também é objeto da superclasse dessa classe.
- III. O objeto da superclasse é um objeto de subclasse da sua classe.
- IV. A palavra-chave *super* é utilizada para que o construtor da subclasse possa chamar o construtor da superclasse.

Estão corretas apenas as afirmativas

- a) I e II.
- b) II e III.
- c) I, III e IV.
- d) I, II e IV.

8. Considere o código abaixo escrito na linguagem Java.

```
1. public class Programa {
2.     private int a;
3.     private int b;
4.     private int c;
5.     private int x;
6.     public void trocar() {
7.         this.b = 4;
8.         this.a = 9;
9.         this.x = 0;
10.        while (x <= 7) {
11.            if ((a % 2) != 2) {
12.                System.out.println(this.a);
13.                this.c = this.b + this.a;
14.                this.b = this.a;
15.                this.a = this.c;
16.            } else {
17.                this.a = this.c;
18.            }
19.        }
20.        this.x++;
21.    }
22. }
23. public static void main(String[] args) {
24.     Programa p = new Programa();
25.     p.trocar();
26. }
27. }
```

Qual é o resultado correto da saída em tela da execução da classe programa?

- a) 9 12 21 33 54 87 141 228
- b) 9 14 23 37 60 97 157 254
- c) 9 13 22 35 56 93 149 241
- d) 9 13 22 35 57 92 149 241

9. Sobre interface na programação orientada a objetos em Java, seguem algumas afirmações:

- I. Uma classe concreta poderá implementar apenas uma interface, utilizando a palavra-chave *implements*.
- II. Uma classe concreta, ao implementar uma interface, estabelece um contrato, escolhendo quais métodos irá implementar em sua estrutura.
- III. As interfaces, além de métodos, podem possuir campos que serão implicitamente finais e estáticos.
- IV. Todos os métodos de uma interface são implicitamente abstratos.

Estão corretas apenas as afirmativas

- a) I e II.
- b) II e III.
- c) III e IV.
- d) I, II e IV.

10. Considere o código escrito na linguagem Java.

```

1. public class Demo {
2.     public static void main(String[] args) {
3.         ArrayList l1=new ArrayList();
4.         ArrayList l2=new ArrayList();
5.         l1.add(2322);
6.         l1.add(3456);
7.         l1.add(1783);
8.         l1.add(2322);
9.         l2.addAll(l1);
10.        TreeSet tr=new TreeSet(l2);
11.        HashSet hs=new HashSet(tr);
12.        System.out.println(tr + " "+hs);
13.    }
14.}

```

Qual é o resultado correto após a execução da classe Demo:

- a) [1783, 2322, 3456] [3456, 2322, 1783]
- b) [3456, 2322, 1783] [3456, 2322, 1783]
- c) [2322, 3456, 1783, 2322] [2322, 3456, 1783, 2322]
- d) [1783, 2322, 2322, 3456] [3456, 2322, 2322, 1783]

Considere o código abaixo para as questões 11 e 12.

<pre> public class Pessoa { public String nome; public String dataN; public Pessoa(String nome, String dataN) { this.nome = nome; this.dataN = dataN; } public Pessoa(){ } public void mostrar() { System.out.println(this.nome + " - " + this.dataN); } } </pre>	<pre> public class Funcionario extends Pessoa{ public int cod; public Funcionario(String nome, String dataN, int cod) { super(nome, dataN); this.cod=cod; } public Funcionario(int cod) { this.cod=cod; } public void mostrar(){ System.out.println("O código do " + "funcionário é: "+this.cod); } } </pre>
---	--

11. Qual o trecho de código abaixo que, se utilizado, gera erro de compilação?

- a) Pessoa p=new Pessoa();
- b) Funcionario f=new Funcionario ();
- c) Funcionario f=new Funcionario(989);
- d) Pessoa f=new Funcionario("Marta", "11/11/1986",545);

12. Sobre as classes *Pessoa* e *Funcionario*, é **INCORRETO** afirmar-se:

- a) Existe o relacionamento tem um.
- b) Para que o método mostrar() da subclasse invoque o método mostrar() da superclasse, é necessário que o método seja precedido por super e um ponto separador(.).
- c) Possuem métodos com a mesma assinatura sem prejudicar sua compilação.
- d) A assinatura @Override poderá ser inserida acima do método mostrar() da subclasse, para indicar que este sobrepõe o método da superclasse.

13. Analise as afirmativas sobre Coleções em Java, assinalando (V) para verdadeiro e (F) para falso.

- () A interface *SortedSet* implementa a interface *Set*.
- () *Vector* é uma classe que implementa a interface *Queue*.
- () A classe *TreeMap* implementa a interface *SortedMap*.
- () Qualquer classe que implementa a interface *Set* sempre terá dados ordenados.
- () A interface *List* estende a interface *Collection*.

A sequência correta, de cima para baixo, é

- a) F-F-F-V-V.
- b) F-F-V-V-V.
- c) V-F-V-F-F.
- d) F-F-V-F-V.

14. Segundo Deitel e Deitel (2010), o tratamento de exceções é utilizado para processar erros síncronos.

Sobre tratamento de exceções no Java, é correto afirmar que relançar exceções ocorre

- a) quando um bloco try é executado.
- b) quando um bloco catch não pode processar uma exceção ou que só pode processá-la parcialmente.
- c) quando uma exceção no bloco try puder ser capturada por um bloco catch.
- d) quando, após uma exceção ser capturada pelo bloco catch, o programa pula e prossegue para o bloco finally.

15. Considere o código abaixo escrito na linguagem PHP.

```

1. <?php
2.     $text="Concurso público";
3.     $busca="o";
4.     $f=0;
5.     $t=strlen($text);
6.     for($i=0; $i<$t; $i++){
7.         if(substr($text, $i, 2)==$busca){
8.             $f=$f+1;
9.         }
10.    }
11.    echo $f;
12. ?>

```

Qual é o resultado exibido em tela após a execução desse código?

- a) 3
- b) 2
- c) 4
- d) 1

16. Observe o código PHP abaixo:

```

1. <?php
2.     $cores=array();
3.     $cores[1]="azul";
4.     $cores["a"]="verde";
5.     $cores[]="amarelo";
6.     $cores[5]="rosa";
7.     $cores[]="roxo";
8.     ksort($cores);
9.     echo key($cores).";";
10.    foreach($cores as $v){
11.    echo $v.";";
12.    }
13. ?>

```

Ao executar o código, qual é a saída exibida em tela?

- a) 2;amarelo;azul;rosa;roxo;verde;
- b) a;verde;azul;amarelo;rosa;roxo;
- c) 1;verde;azul;amarelo;rosa;roxo;
- d) azul;amarelo;verde;rosa;roxo;1;

17. Analise o código abaixo.

```

class Teste {
    public $nome;
    public $idade;
}

```

Qual é o trecho correto de código escrito na linguagem PHP, orientada a objetos, que insere o construtor na classe *Teste*?

- a) public function __construct(\$nome, \$idade) {
 \$this->nome = \$nome;
 \$this->idade = \$idade;
 }
- b) public function construct(\$nome, \$idade) {
 this->\$nome = \$nome;
 this->\$idade = \$idade;
 }
- c) public function __Teste(\$nome, \$idade) {
 \$this->nome = \$nome;
 \$this->idade = \$idade;
 }
- d) public function Teste(\$nome, \$idade) {
 this->\$nome = \$nome;
 this->\$idade = \$idade;
 }

18. São diretivas na codificação com JSP e utilizadas para informações especiais em uma página, respectivamente:

- a) @out, @request, @session.
- b) @include, @page, @taglib.
- c) @config, @response, @out.
- d) @application, @print, @type.

19.De acordo com Duckett (2010), no HTML existem elementos de apresentação que possuem opções para indicar como um texto pode aparecer em HTML e XHTML. Também são encontrados outros elementos que representam expressão, e o uso desses elementos adiciona mais informações a um documento. Dentre os elementos de expressão, o elemento _____ é usado para a Amostra do Programa, indicando saída de exemplo de um programa, script ou algo semelhante.

Qual é o elemento de expressão que preenche corretamente a lacuna acima?

- a) <samp>
- b) <dfn>
- c) <blockquote>
- d) <abbr>

20.Em HTML, utiliza-se âncora de destino, com os atributos *name* e *id*, para

- a) transferir arquivos.
- b) fazer uma conexão e abrir o editor de e-mails padrão do usuário.
- c) marcar pontos específicos de uma página HTML, a qual a conexão pode apontar.
- d) conectar-se a outro documento, onde o valor dos atributos *name* ou *id* possui a página a que está se conectando.

21.Tabelas em HTML, segundo Duckett (2010), são utilizadas para apresentar dados em forma de planilhas constituídas de linhas e colunas.

Sobre tabelas, é **INCORRETO** afirmar que

- a) uma tabela pode ser dividida em três seções: cabeçalho, corpo e rodapé.
- b) as células podem ocupar mais de uma linha ou coluna.
- c) o atributo *colspan* é utilizado para agrupar uma ou mais colunas adjacentes.
- d) o elemento <caption> é utilizado para adicionar um texto em uma tabela.

22.Qual caractere representa que uma regra CSS (*Cascading Style Sheets*) se aplica a todos os elementos HTML?

- a) \$
- b) *
- c) .
- d) #

23. Analise o código abaixo.

```
1. <html>
2.   <head>
3.     <style>
4.       div+p{
5.         background-color: yellow;
6.       }
7.     </style>
8.   </head>
9.   <body>
10.    <h1> CSS</h1>
11.    <div>
12.      <h2>Vamos aprender CSS</h2>
13.      <p>Primeiro item:seletores</p>
14.      <p>Segundo item:propriedades</p>
15.    </div>
16.    <p>CSS permite associar regras aos elementos</p>
17.  </body>
18.</html>
```

Qual(ais) linha(s) o seletor definido na linha 5 afetará?

- a) Linhas 13 e 14.
- b) Linhas 13, 14 e 16.
- c) Apenas a linha 13.
- d) Apenas a linha 16.

24. O evento jQuery _____ é inserido para que todo o script ali contido só entre em ação depois que o documento estiver carregado.

Qual é a opção que completa corretamente a lacuna acima?

- a) `$(document).ready()`
- b) `$(document).show()`
- c) `$(document).loading()`
- d) `$(document).down()`

25. Observe o trecho de código jQuery abaixo.

```
$('#input').bind('blur', function(){
    $(this).addClass('destaque');
});
```

O que acontecerá se esse trecho for inserido em uma página HTML?

- a) Irá gerar um erro, pois `.bind()` não existe no jQuery.
- b) Selecionará todos os elementos `input`, adicionando um evento de perda de foco e relacionando-os a uma classe chamada `destaque`.
- c) Selecionará todos os elementos `input`, adicionando um evento de ganho de foco e relacionando-os a uma classe chamada `destaque`.
- d) Selecionará apenas o primeiro elemento `input` de uma página, removerá um evento de perda de foco e, em seguida, adicionará uma classe chamada `destaque`.

26. Considere o trecho de código Android a seguir.

```
x = getIntent().getExtras();
x.putSerializable("usuario", new Usuario(1, "Marlene"));
```

O tipo do objeto *x* é

- a) Bundle.
- b) Extras.
- c) ArrayList.
- d) ArrayAdapter.

27. Classe utilizada para que aplicações possam reagir a eventos gerados por uma *Intent*, sendo executada em segundo plano durante pouco tempo e sem utilizar uma interface gráfica.

A definição acima corresponde a

- a) AlarmManager.
- b) BroadcastReceiver.
- c) Service.
- d) Notification.

28. Considere as seguintes afirmações sobre o ciclo de vida de uma *Activity*, e assinale (**V**) para verdadeiro e (**F**) para falso:

- () O método *onCreate(bundle)* é obrigatório e é chamado uma única vez durante todo o ciclo de vida de uma *Activity*.
- () O método *onStart()* é chamado quando a *Activity* está no topo da pilha de execução.
- () Em casos de falta de memória, o sistema operacional do Android pode decidir encerrar uma *Activity* que esteja parada em segundo plano.
- () Quando uma *Activity* está em execução e uma ligação é recebida e atendida pelo usuário, o método *onStop()* é chamado.
- () O método *onDestroy()* é executado quando o método *finish()* da classe *Activity* é chamado pela aplicação.

A sequência correta, de cima para baixo, é

- a) V - F - V - F - F.
- b) F - V - F - V - F.
- c) F - F - V - V - V.
- d) V - F - V - F - V.

29. Com relação aos gerenciadores de layout no Android, afirma-se que

- I. o *LinearLayout* permite organizar componentes na horizontal e vertical.
- II. o *RelativeLayout* permite organizar componentes através de coordenadas x e y.
- III. a classe *ViewGroup* herda da classe *View* e é classe-mãe das classes *FrameLayout*, *LinearLayout* e *TableLayout*.
- IV. um *LinearLayout* com a propriedade *android:layout_width="match_parent"* significa que ele irá ocupar o tamanho, em largura, necessário na tela.

Está(ão) correta(s) apenas a(s) afirmativa(s)

- a) I e II.
- b) II e IV.
- c) I e III.
- d) I, III e IV.

30. Com relação aos componentes de interface gráfica do Android, são feitas as seguintes afirmações:

- I. Através do método *setOnSelectedListener(OnSelectedListener)*, é possível verificar quando um *CheckBox* é selecionado.
- II. A propriedade *android:completionThreshold* do *AutoCompleteTextView* define o número mínimo de caracteres que o usuário precisa digitar para iniciar o autopreenchimento do texto.
- III. Através do método *setOnItemSelectedListener(OnItemSelectedListener)*, é possível adicionar uma ação em um *Spinner* e verificar qual item foi selecionado pelo usuário.
- IV. É possível preencher as opções de um *Spinner* através de um objeto do tipo *ArrayAdapter<Usuario>* com o método *setListAdapter(ArrayAdapter)*.

Estão corretas apenas as afirmativas

- a) I e II.
- b) I e IV.
- c) II e III.
- d) II, III e IV.

31. Com relação à transformação de modelos Entidade-Relacionamento (ER) em modelos relacionais, são feitas as seguintes afirmações:

- I. Em relacionamentos 1:1, em que uma das entidades têm participação obrigatória e a outra entidade tem participação opcional, a tradução preferida é através da fusão das tabelas correspondentes às duas entidades.
- II. Em relacionamentos 1:1, em que ambas entidades têm participação obrigatória, a tradução preferida é através da criação de uma tabela própria.
- III. Independentemente da cardinalidade mínima, relacionamentos n:n são sempre implementados através de uma tabela própria.
- IV. Em relacionamentos 1:n, a alternativa preferida de implementação é a de adição de colunas. No entanto, em relacionamentos 1:n, em que a entidade com cardinalidade máxima é opcional, pode ser usada, alternativamente, a implementação através de tabela própria.

Está(ão) correta(s) apenas a(s) afirmativa(s)

- a) I e II.
- b) I e III.
- c) II e IV.
- d) I, III e IV.

32. Qual das seguintes operações em subconsultas é equivalente à operação IN?

- a) >ANY
- b) =ANY
- c) <ANY
- d) >ALL

33. Observe o seguinte código.

```
CREATE FUNCTION funcao(text, VARIADIC anyarray) RETURNS text AS $$  
SELECT array_to_string($2, $1);  
$$ LANGUAGE SQL;  
SELECT funcao('1',2,3,4);
```

Qual é o resultado da execução desse código?

- a) 4,3,2,1
- b) 1234
- c) 21314
- d) 1,2,3,4

34. Considere a tabela Pessoa com as seguintes informações:

Pessoa		
id	nome	data_nasc
1	Maria	2017-10-09
2	João	2019-11-14
3	Ana	2016-10-14
4	Pedro	2017-11-09

Considere também a seguinte instrução SQL (*Structured Query Language*):

```
SELECT nome FROM Pessoa WHERE data_nasc IN (SELECT data_nasc FROM (SELECT  
data_nasc FROM (SELECT data_nasc FROM Pessoa WHERE EXTRACT(YEAR FROM data_nasc) =  
(EXTRACT(YEAR FROM NOW())))) AS nasc WHERE EXTRACT(MONTH FROM data_nasc) =  
(EXTRACT(MONTH FROM NOW())))) AS nasc WHERE EXTRACT(DAY FROM data_nasc) <= 10);
```

Qual é o nome que resultará da execução dessa consulta, caso tenha sido executada em 20/11/2017?

- a) Ana.
- b) João.
- c) Maria.
- d) Pedro.

35.Considere as seguintes tabelas:

Cliente	
cod_cliente	nome
1	Rodrigo
2	Carlos
3	Rosana

Vendedor	
cod_vendedor	nome
1	Verônica
2	Cristian
3	Renan

Pedido		
cod_pedido	cod_vendedor	cod_cliente
1	1	1

A instrução SQL que retorna as informações representada em ?

Vendedor	Cliente
Carlos	
Rodrigo	Verônica
Rosana	
	Cristian
	Renan

está corretamente

- a) `SELECT Cliente, Vendedor FROM ((SELECT c.nome as Cliente, v.nome as Vendedor FROM Cliente c LEFT JOIN Pedido p ON p.cod_cliente = c.cod_cliente LEFT JOIN Vendedor v ON v.cod_vendedor = p.cod_vendedor) UNION (SELECT c.nome as Cliente, v.nome as Vendedor FROM Cliente c RIGHT JOIN Pedido p ON p.cod_cliente = c.cod_cliente RIGHT JOIN Vendedor v ON v.cod_vendedor = p.cod_vendedor)) as temp ORDER BY Cliente ASC, Vendedor ASC;`
- b) `SELECT Cliente, Vendedor FROM ((SELECT c.nome as Cliente, v.nome as Vendedor FROM Cliente c RIGHT JOIN Pedido p ON p.cod_cliente = c.cod_cliente RIGHT JOIN Vendedor v ON v.cod_vendedor = p.cod_vendedor) UNION (SELECT c.nome as Cliente, v.nome as Vendedor FROM Cliente c RIGHT JOIN Pedido p ON p.cod_cliente = c.cod_cliente RIGHT JOIN Vendedor v ON v.cod_vendedor = p.cod_vendedor)) as temp ORDER BY Cliente ASC, Vendedor ASC;`
- c) `SELECT Cliente, Vendedor FROM ((SELECT c.nome AS Cliente, v.nome AS Vendedor FROM Cliente c LEFT JOIN Pedido p ON p.cod_cliente = c.cod_cliente LEFT JOIN Vendedor v ON v.cod_vendedor = p.cod_vendedor) UNION (SELECT c.nome AS Cliente, v.nome AS Vendedor FROM Cliente c LEFT JOIN Pedido p ON p.cod_cliente = c.cod_cliente LEFT JOIN Vendedor v ON v.cod_vendedor = p.cod_vendedor)) AS temp ORDER BY Cliente ASC, Vendedor ASC;`
- d) `SELECT Cliente, Vendedor FROM ((SELECT c.nome as Cliente, v.nome as Vendedor FROM Cliente c JOIN Pedido p ON p.cod_cliente = c.cod_cliente JOIN Vendedor v ON v.cod_vendedor = p.cod_vendedor) UNION (SELECT c.nome as Cliente, v.nome as Vendedor FROM Cliente c JOIN Pedido p ON p.cod_cliente = c.cod_cliente JOIN Vendedor v ON v.cod_vendedor = p.cod_vendedor ORDER BY v.nome)) as temp ORDER BY Cliente ASC, Vendedor ASC;`

36.Um multiprocessador de memória compartilhada é um sistema de computador no qual duas ou mais CPUs compartilham acesso total a uma memória RAM comum. São tipos de sistemas operacionais utilizados em multiprocessadores:

- a) Multiprocessadores "Cliente-servidor".
- b) Multiprocessadores Simétricos.
- c) Multiprocessadores NUMA.
- d) Multiprocessadores Fracamente Acoplados.

37.Quando um computador é multiprogramado, ele muitas vezes tem múltiplos processos ou threads que competem pela CPU ao mesmo tempo. Se somente uma CPU se encontrar disponível, deverá ser feita uma escolha de qual processo executará em seguida.

Qual parte do sistema operacional fará esta escolha?

- a) Algoritmo FIFO.
- b) Algoritmo LIFO.
- c) Algoritmo de Escalonamento.
- d) Algoritmo de Melhor Esforço.

38.Com relação às permissões especiais utilizadas nos arquivos do sistema operacional Linux, pode-se utilizar por questões de segurança os recursos SUID e o SGID.

Qual é o efeito da utilização do SUID em arquivos executáveis?

- a) Quem executar o arquivo terá os mesmos direitos e poderes do dono do arquivo, durante a sua execução.
- b) Atualmente, nenhum.
- c) Todos os arquivos e diretórios criados pertencerão ao mesmo grupo do diretório que os engloba.
- d) Quem executar o arquivo terá os mesmos direitos e poderes do grupo do arquivo, durante a sua execução.

39.A respeito de comandos básicos utilizados no sistema operacional Linux, é **INCORRETO** afirmar que o comando

- a) `wc -l /etc/passwd` mostra a quantidade de linhas no arquivo `/etc/passwd`.
- b) `ls` mostra uma lista de arquivos.
- c) `free -s 1` mostra a utilização da memória a cada 1 segundo.
- d) `touch -c ifsul` muda a data e hora do arquivo `ifsul` independente de ele existir ou não.

40.Sobre as funções básicas da gerência de memória em um sistema operacional, afirma-se que

- I. permite a execução de programas maiores que a memória física disponível.
- II. protege as áreas de memória ocupadas por cada processo.
- III. monitora todos os acessos ao barramento da máquina.
- IV. mantém na memória principal o maior número de processos residentes.

Estão corretas apenas as afirmativas.

- a) I e II.
- b) II, III e IV.
- c) I, III e IV.
- d) I, II e IV

