-- CONHECIMENTOS ESPECÍFICOS --

```
1  k ← 0
2  para i ← 0, 1, ..., n-1 faça
3      se P(A[i]) então
4          A[k] ← A[i]
5          k ← k + 1
6      fim se
7  fim para
```

No pseudocódigo precedente, o vetor `A` de tamanho $n \geq 1$ é indexado de 0 a $n - 1$; o predicado booleano `P` é puro, fixo e depende exclusivamente do valor recebido; o vetor `A_original` tem o mesmo conteúdo de `A` antes da execução de qualquer linha do pseudocódigo em questão.

A partir dessas informações, julgue o item a seguir.

51 Ao fim da execução do algoritmo, as primeiras `k` posições de `A` conterão somente as ocorrências de `A_original` para as quais `P` é verdadeiro, na mesma ordem relativa em que apareciam no vetor original, e, ainda, se `k = 0`, esse prefixo será vazio.

Acerca dos tipos de dados simples e estruturados, julgue os itens seguintes.

52 Considerando-se que, no pseudocódigo a seguir, o comando de entrada `leia()` atribui o primeiro valor fornecido a `p.primeiro` e o segundo a `p.segundo`, e admitindo-se aritmética inteira exata, é correto afirmar que o comando de saída `escreva()` apresentará (`p.primeiro – p.segundo`) e `p.primeiro`, nessa ordem.

```
1  tipo Par = registro
2      primeiro: inteiro
3      segundo: inteiro
4  fim registro
5  variável p: Par
6  leia(p.primeiro, p.segundo)
7  p.primeiro ← p.primeiro + p.segundo
8  p.segundo ← p.primeiro - p.segundo
9  p.primeiro ← p.primeiro - p.segundo
10 escreva(p.primeiro, p.segundo)
```

53 No código a seguir, `catalogo` é uma variável de tipo estruturado homogêneo, embora tenha a dimensão definida pela constante `N` e armazene registros com campos de tipos distintos.

```
1  constante N: inteiro ← 50
2  tipo Item = registro
3      codigo: inteiro
4      preco: real
5      ativo: lógico
6  fim registro
7  variável catalogo: vetor[0..N-1] de Item
```

Julgue os itens que se seguem, referentes a avaliação de expressões e funções pré-definidas.

54 Considerando-se que a função `piso(x)` retorna o maior número inteiro menor ou igual a `x` e que a função `trunca(y)` retorna a parte inteira de um número `y`, então, para todo número `z` real, negativo e não inteiro, `piso(z) = trunca(z) - 1` será válido.

55 Considere que, no pseudocódigo a seguir, seja adotada avaliação de curto-circuito para o operador lógico `e`, de modo que o segundo operando somente seja avaliado quando o primeiro for verdadeiro.

```
1  X ← 0
2  Resultado ← (x ≠ 0) e ((10 div x) > 2)
```

Nessa situação, a precedência dos operadores `div` e `>` determina a avaliação de `(10 div x) > 2` antes da conjunção lógica, provocando uma divisão por zero.

A respeito de bloco de comandos, estruturas de controle, seleção, repetição e desvio, e operadores e expressões, julgue os próximos itens.

56 Ao fim da execução do pseudocódigo apresentado a seguir, a função `escreva(contador)` apresentará o valor 4.

```
1  contador ← 0
2  para i ← 1 até i ≤ 3 faça
3      para j ← 1 até i ≤ 3 faça
4          se i = j então
5              continue
6          fim se
7          se i + j > 4 então
8              interrompa
9          fim se
10         contador ← contador + 1
11     fim para
12 fim para
13 escreva(contador)
```

57 Considerando-se o pseudocódigo a seguir, é correto afirmar que, após a execução do bloco interno, a função `escreva(x)` apresentará o valor `3`.

```
1  variável x: inteiro
2  x ← 4
3  início
4      variável x: inteiro
5      variável y: inteiro
6      y ← 5
7      x ← y - 2
8  fim
9  escreva(x)
```

58 Se o operador `div` representar a parte inteira do quociente de uma divisão para todos os inteiros não negativos `a` e `b` e todo inteiro positivo `c`, então a expressão a seguir será verdadeira.
`(a + b) div c = (a div c) + (b div c)`

Julgue os itens subsequentes, que dizem respeito a passagem de parâmetros, recursividade, noções de programação estruturada e orientada a objetos e métodos de ordenação, de pesquisa e de *hashing*.

59 Considere que, no pseudocódigo a seguir, o parâmetro `ref x` constitua uma referência à variável fornecida como argumento, enquanto `valor y` receba uma cópia independente.

```
1  procedimento ajustar(ref x: inteiro,
   valor y: inteiro)
2      x ← x + y
3      y ← x - y
4      x ← x + y
5  fim procedimento
6  variáveis a, b: inteiro
7  a ← 4
8  b ← 3
9  ajustar(a, b)
10 escreva(a, b)
```

Com base nessa situação hipotética e admitindo-se aritmética inteira exata, é correto afirmar que, ao término da execução, o comando `escreva(a, b)` apresentará os valores `11` e `3`, nessa ordem.

60 No pseudocódigo a seguir, a função `fatorial` não é recursiva de cauda, pois a multiplicação `n*acumulador` permanece pendente até o retorno da chamada recursiva.

```
1  função fatorial(n: inteiro, acumulador:
   inteiro): inteiro
2      se n ≤ 1 então
3          retorne acumulador
4      fim se
5      retorne fatorial(n - 1,
   n*acumulador)
6  fim função
```

61 Considere o modelo orientado a objetos em pseudocódigo a seguir, em que é empregado despacho dinâmico para métodos sobrescritos.

```
1  classe  Processador
2      método calcular(): inteiro
3          retorne 0
4      fim método
5  fim classe
6  classe ProcessadorLote herda
   Processador
7      sobrescreve método calcular():
   inteiro
8          variável total: inteiro
9          total ← 0
10         para i ← 1 até i ≤ 3 faça
11             total ← total + i
12         fim para
13         retorne total
14     fim método
15 fim classe
16 variável p: Processador
17 p ← novo ProcessadorLote()
18 escreva(p.calcular())
```

Nessa implementação, a chamada de `calcular()` executará o método de `ProcessadorLote`, cujo bloco estruturado fará o comando de saída apresentar o valor `6`.

62 Na ordenação crescente de um vetor pelo método *insertion sort*, imediatamente antes de cada iteração do laço externo, o trecho já processado encontra-se ordenado e contém os menores elementos de todo o vetor, embora a parte restante ainda possa estar desordenada.

63 Considere que uma tabela *hash* inicialmente vazia apresente posições numeradas de 0 a 6, a função `h(k) = k mod 7` e tratamento de colisões por endereçamento aberto com sondagem linear, cuja sequência é `(h(k) + i) mod 7`, para `i` = 0, 1, 2, ... Considere, também, que as chaves `10`, `17` e `24` sejam inseridas nessa ordem. Nesse caso, uma pesquisa posterior pela chave `24` examinará, nessa ordem, as posições `3`, `4` e `5`, e encontrará a referida chave na última posição.

Julgue os próximos itens, relativos a arquitetura de aplicações, padrão arquitetural MVC (*model-view-controller*) e arquitetura de microsserviços.

64 A preservação dos limites dos contextos delimitados (*bounded contexts*) e a busca pelo elevado desacoplamento justificam a preferência pela duplicação de código em vez do reúso via herança ou classes compartilhadas.

65 Para que se obtenha granularidade mais próxima possível do ideal na arquitetura de microsserviços, os serviços devem ser projetados no menor tamanho possível, admitindo-se o compartilhamento de bancos de dados para otimizar o processamento das chamadas de rede.

66 Na programação orientada a objetos, a métrica LCOM (*lack of cohesion in methods*) é empregada para mensurar a ausência de coesão de uma classe, de forma que, nessa métrica, valores elevados indicam baixa coesão estrutural, sinalizando que os métodos operam sobre conjuntos distintos de atributos.

67 Uma vez que a conascência dinâmica tem maior força que a estática, os processos de refatoração devem priorizar a transformação de dependências dinâmicas em formas estáticas de menor força, como a conascência de nome.

68 No padrão MVC, a notificação de mudança de estado emitida pelo modelo é enviada ao controlador, que repassa os novos dados à camada visão (*view*) para a atualização da interface.

69 No padrão arquitetural MVC, as regras de negócio do sistema são implementadas diretamente na camada visão (*view*).

Acerca de arquitetura orientada a eventos, de refatoração, de arquitetura *cloud native* e de padrões de *design* de *software*, julgue os itens subsequentes.

70 Na abordagem *cloud native*, a resiliência pressupõe que falhas em sistemas distribuídos são inevitáveis, orientando o projeto arquitetural para a contenção de danos e a recuperação graciosa, por meio de estratégias como *circuit breakers* em vez da tentativa de eliminação completa dos erros.

71 O padrão criacional Singleton permite que uma classe tenha diversas instâncias no sistema, disponibilizando um ponto de acesso global para cada uma delas.

72 Em sistemas cuja estrutura esteja severamente degradada, a refatoração isolada do código é insuficiente para corrigir o problema, sendo necessária a refatoração de projeto para substituir o código existente por padrões de projeto.

73 Em arquiteturas orientadas a eventos que utilizam a topologia do *broker*, a comunicação baseia-se na emissão de comandos, o que garante o controle centralizado do fluxo e facilita a recuperação de transações em caso de falha.

A respeito de interoperabilidade entre aplicações, de API *gateway*, de servidores de aplicações e de conteinerização de aplicação, julgue os itens subsecutivos.

74 O isolamento de processos no Docker por meio de PID *namespace* permite que um mesmo processo possua identificadores (PIDs) distintos dentro do contêiner e no sistema operacional hospedeiro.

75 Em um sistema de caixas eletrônicos projetado na arquitetura cliente-servidor de duas camadas com cliente gordo, a comunicação dos terminais com a base de dados central requer um monitor de teleprocessamento como *middleware*, encarregado de dispor as transações em série para evitar a interferência mútua e garantir a integridade do banco de dados.

76 Em arquiteturas orientadas a mensagens, o padrão Content Enricher é utilizado para complementar uma mensagem com informações obtidas de uma fonte externa antes que ela seja entregue ao sistema de destino.

77 A utilização de uma aplicação intermediária para unificar a suíte de serviços em um sistema de negociação de ações é desaconselhada porque essa arquitetura impede a coordenação entre os diferentes sistemas de *back-end*.

78 O API *gateway* impede a agregação de respostas, de modo que a aplicação cliente precisa realizar requisições individuais a cada serviço de *back-end* para obter as informações necessárias.

79 Nas arquiteturas de microsserviços, o API *gateway* é responsável pelo gerenciamento do tráfego norte-sul, que consiste no roteamento das requisições de clientes externos para os serviços de *back-end*, cabendo às malhas de serviço (*service mesh*) o tratamento do tráfego leste-oeste, referente à comunicação interna entre os serviços.

Em relação a conceitos e ferramentas de DevOps, técnicas de integração e implantação CI/CD, gerência de configuração de *software* (GIT), *web standards* (W3C) e autenticação única (*single sign-on*), julgue os próximos itens.

80 No OAuth 2.0, ao utilizar o fluxo de concessão por asserção (*assertion grant type*), o cliente fica dispensado de autenticar a própria identidade no *endpoint* de *token*, desde que a asserção enviada via requisição HTTP POST seja criptograficamente válida.

81 De acordo com as diretrizes da WCAG 2, a exigência de que todas as funcionalidades de uma aplicação *web* sejam acessíveis por meio da navegação exclusivamente via teclado refere-se ao princípio perceptível.

82 Na integração com o gerenciamento de serviços de TI, o DevOps atua sob a lógica *pull*, enquanto o ITSM atua sob a lógica *push*.

83 No ciclo de vida do DevOps, ferramentas como o Git e o Gerrit são utilizadas na fase de codificação (*code*).

84 No sistema de controle de versão Git, os ramos (*branches*) funcionam essencialmente como ponteiros móveis que indicam *commits* específicos no repositório.

85 A ocorrência do *integration hell* constitui um efeito colateral da integração contínua (CI), decorrente do excesso de *commits* e mesclagens diárias efetuados pelos desenvolvedores no repositório central.

A respeito de banco de dados, julgue os próximos itens.

86 O modelo lógico de bancos de dados apresenta o conceito de chaves primárias e estrangeiras, incluindo a definição exata de colunas, índices e partições.

87 A junção de duas consultas pelo operador `UNION` requer que ambas tenham a mesma quantidade de colunas e tipos de dados compatíveis nas mesmas posições.

88 O modelo conceitual de banco de dados é independente de tecnologia RDBMS e foca a representação gráfica das regras de negócio por meio de entidades e relacionamentos.

89 O *tablespace* chamado `SYSTEM` é opcional em bancos de dados Oracle e pode armazenar os dados do dicionário de dados do sistema.

90 O otimizador de consultas pode considerar um *full table scan* mais rápido que o uso de índices se o resultado apresentar um percentual muito elevado dos blocos da tabela.

Acerca de linguagens de programação, julgue os itens que se seguem.

91 Em Javascript, constantes declaradas com `const` não podem ser reatribuídas, embora suas propriedades internas possam ser modificadas.

92 A verificação de tipos na linguagem Java é considerada dinamicamente tipada e pode ocorrer em tempo de compilação ou de execução.

93 A linguagem Java utiliza passagem de parâmetro por valor, permitindo alteração do endereço de memória dos ponteiros originais das variáveis.

94 Na linguagem Python, variáveis dentro de funções podem acessar variáveis externas de escopo global, mas a modificação só ocorre com utilização da palavra-chave `global`.

Com relação a tecnologias e práticas *frontend web*, julgue os itens a seguir.

95 Aplicações PWA podem ser instaladas na tela inicial do dispositivo móvel mesmo sem a existência do arquivo manifest.json configurado.

96 Os dados em requisições Ajax assíncronas podem ser transferidos por meio de API específica sem necessidade de recarregar a página inteira.

97 Considerados os estados internos de componentes de *frontend*, é correto afirmar que, no React, os *hooks* podem ser chamados no nível superior das funções ou dentro de condicionais e laços de repetição.

98 Aplicações SPA (*single page application*) carregam um documento HTML inicial e atualizam dinamicamente a página, utilizando DOM via Javascript.

No que tange a protocolos HTTPS e SSL/TLS, julgue os itens subsequentes.

99 A fase inicial do protocolo TLS, conhecido como *handshake,* utiliza conexão com criptografia simétrica, com o objetivo de autenticar o servidor e trocar as chaves de forma segura.

100 O protocolo HTTPS encapsula as mensagens HTTP em uma camada criptográfica, visando garantir confidencialidade e integridade na comunicação entre cliente e servidor.

Acerca de disponibilidade, autorização, classificação da informação, segurança física e segurança lógica, julgue os itens a seguir.

101 A utilização de mecanismos de controle de acesso lógico permite que se restrinja o acesso a recursos de um sistema de informação com base nas credenciais e nas permissões atribuídas aos usuários.

102 A rotulagem da informação corresponde ao processo por meio do qual o nível de classificação da informação é estabelecido para definir seu tratamento de segurança.

103 A disponibilidade da informação está relacionada à preservação dos dados contra alterações não autorizadas, de modo que os usuários autorizados possam acessá-los quando necessário.

104 Na autorização baseada em funções, as permissões de acesso são atribuídas diretamente aos usuários que exercem determinada função.

105 A classificação da informação orienta a definição dos controles de proteção aplicáveis, com base nos impactos associados à perda de confidencialidade, integridade ou disponibilidade da informação.

106 A instalação de câmeras de videomonitoramento nos acessos de entrada de uma organização constitui um controle de entrada física destinado a impedir o acesso não autorizado às instalações.

Julgue os itens seguintes, a respeito de sistemas criptográficos simétricos e assimétricos, *hash*, protocolos criptográficos e infraestrutura de chaves públicas – ICP-Brasil.

107 Durante a preservação de um arquivo, a alteração de atributos mantidos pelo sistema de arquivos, como permissões de acesso e *timestamps*, não altera o valor de *hash* previamente calculado sobre os *bytes* do arquivo.

108 Durante o estabelecimento de uma conexão TLS 1.3, as chaves utilizadas para proteger os dados da aplicação são derivadas diretamente da chave pública contida no certificado digital apresentado pelo servidor.

109 Em sistemas criptográficos simétricos com cifras de bloco em modos de operação que empregam vetor de inicialização, a utilização adequada desse vetor permite ao destinatário verificar a integridade do conteúdo cifrado após a decifragem.

110 Em sistemas criptográficos assimétricos, a cifragem de uma mensagem com a chave pública do destinatário garante a identificação da autoria da mensagem após sua decifragem.

111 Em uma aplicação que valida uma assinatura digital ICP-Brasil, a verificação do certificado do signatário inclui tanto a análise de sua cadeia de certificação quanto um certificado de confiança da ICP-Brasil.

Julgue os itens subsequentes, em relação a controle de acesso e vulnerabilidades de aplicações *web*.

112 Em uma aplicação *web* sujeita a um ataque de fixação de sessão, a regeneração do identificador de sessão após a autenticação do usuário reduz o risco de que um identificador conhecido pelo atacante permaneça associado à sessão autenticada.

113 Em um sistema de controle de acesso em que se utilizam cartões de acesso como fator de posse e senhas pessoais como fator de conhecimento, a utilização conjunta desses elementos no processo caracteriza autenticação multifator.

114 Em um sistema biométrico de controle de acesso, a alteração do limiar de decisão modifica o critério de aceitação da correspondência biométrica e afeta as taxas de falsa aceitação e de falsa rejeição.

115 Em uma aplicação *web,* o uso de consultas SQL parametrizadas como medida de prevenção à injeção permite que os identificadores de tabelas e colunas fornecidos pelo usuário sejam tratados da mesma forma que os valores utilizados nos filtros.

No que diz respeito a segurança na comunicação pela Internet, engenharia social e noções de privacidade e proteção de dados segundo a Lei Geral de Proteção de Dados Pessoais (LGPD), julgue os itens subsecutivos.

116 Uma pessoa pode exercer os direitos assegurados pela LGPD em relação ao tratamento de seus dados pessoais ainda que o tratamento realizado pelo controlador esteja fundamentado no cumprimento de obrigação legal ou regulatória.

117 Em ataques de engenharia social, o conhecimento prévio acerca de processos, relacionamentos ou características de uma organização pode ser utilizado para a construção de uma abordagem coerente com o contexto da vítima e aumentar a probabilidade de que ela realize a ação pretendida pelo atacante.

118 Em um acesso remoto a um servidor HTTPS por meio de uma VPN, esta protege o tráfego entre os *endpoints* do túnel, enquanto o TLS utilizado pelo HTTPS protege criptograficamente a comunicação entre os *endpoints* da aplicação.

119 Uma empresa contratada para realizar o tratamento de dados pessoais em nome de outra empresa que figure como controladora desses dados passa a ser considerada controladora visto que executa diretamente as operações de tratamento.

120 A obtenção de um dado pessoal sensível a partir de dados cujo acesso seja público faz que ele deixe de ser considerado sensível para fins de aplicação da LGPD.

Espaço livre