

MESTRADO PROFISSIONAL EM ENGENHARIA DE CONTROLE E AUTOMAÇÃO SISTEMAS INTELIGENTES

LEIA ATENTAMENTE AS INSTRUÇÕES ABAIXO

1. Você recebeu do fiscal o seguinte material:

- (a) Este caderno, com o enunciado das 40 (quarenta) questões objetivas, sem repetições ou falhas, com a seguinte distribuição:

Língua Inglesa	Conhecimentos Específicos
Questões de 1 a 20	Questões de 21 a 40

- (b) O CARTÃO-RESPOSTA destinado às respostas das questões objetivas formuladas nas provas.

2. Todas as questões valem 5 (cinco) pontos. Assim, as provas de Língua Inglesa (questões de 1 a 20) e de Conhecimentos Específicos (questões de 21 a 40) valem 100 (cem) pontos cada.
3. Após a conferência, o candidato deverá assinar, no espaço próprio do CARTÃO-RESPOSTA, a caneta esferográfica na cor azul ou preta.
4. Para cada uma das questões objetivas, são apresentadas 5 alternativas classificadas com as letras (a), (b), (c) e (d); só uma responde adequadamente ao quesito proposto. Você só deve assinalar UMA RESPOSTA: a marcação em mais de uma alternativa anula a questão, MESMO QUE UMA DAS RESPOSTAS ESTEJA CORRETA.
5. SERÁ ELIMINADO do Processo Seletivo Público o candidato que:
- (a) Se utilizar, durante a realização das provas, de máquinas e/ou relógios de calcular, bem como de rádios gravadores, *headphones*, telefones celulares ou fontes de consulta de qualquer espécie;
- (b) Se ausentar da sala em que se realizam as provas levando consigo o CADERNO DE QUESTÕES e/ou o CARTÃO-RESPOSTA.
- (c) Não assinar a LISTA DE PRESENÇA e/ou o CARTÃO-RESPOSTA.
- Obs.: O candidato só poderá se ausentar do recinto das provas após 1 (uma) hora contada a partir do efetivo início das mesmas. Por motivos de segurança, o candidato só poderá levar o CADERNO DE QUESTÕES, depois de 3 (três) horas contadas a partir de efetivo início da prova.

6. Reserve os 30 (trinta) minutos finais para marcar seu CARTÃO-RESPOSTA.

7. Quando terminar, entregue ao fiscal, o CARTÃO-RESPOSTA e ASSINE A LISTA DE PRESENÇA.

8. O TEMPO DISPONÍVEL PARA ESTAS PROVAS DE QUESTÕES OBJETIVAS É DE 5 (CINCO) HORAS, incluído o tempo para a marcação do seu CARTÃO-RESPOSTA.

PROVA DE INGLÊS

Texto 1: **Eliminating Indecision About the Internet of Things**

- 1 If you've been seriously exploring an Internet of Things (IoT) initiative for some aspect of your operations, you've
2 encountered OPC UA (Unified Architecture) and DDS (Data Distribution Service). These two data communication
3 technologies appear, at first glance, to accomplish many of the same tasks, which has led to quite a bit of concern
4 and confusion in an industry weary from the fieldbus and Ethernet protocol wars.
5 The OPC UA and DDS technologies seemed to be yet another instance of competing approaches that would
6 force users to choose one approach over another. For most users, the uncertainty about which one will eventually
7 become the preferred industry standard causes them to sit on the sidelines to see which prevails — which can be
8 a dangerous business strategy considering the speed at which the Internet of Things is changing business.
9 In an effort to remove this decision hurdle for industrial companies, the OPC Foundation (which supports OPC
10 UA) and the Object Management Group (which supports DDS) have announced an agreement to develop an "OPC
11 UA/DDS gateway" specification that will permit independent implementations of each technology to work together
12 more easily, as well as an "OPC UA DDS Profile" for integrated use cases.

As questões de 1 a 10 são referentes ao Texto 1.

- The pronoun "These" (line 2) refers to:
 - Different aspects of the Internet of Things.
 - Unified Architecture.
 - Data Distribution Service.
 - Different data communication technologies.
- The word "weary" (line 4) could be replaced, without changing the meaning of the text, by:
 - weakened
 - unused
 - fatigued
 - invigorated
- Which of the options below contains an idea presented in the first paragraph?
 - Both UA and DDS are technologies that complement one another.
 - Both UA and DDS will solve the problem presented by the Internet of Things.
 - Both UA and DDS have led to preoccupation and misunderstanding in the industry.
 - Both UA and DDS are outdated data communication technologies.
- The pronoun "that" (line 5) can be replaced by:
 - who
 - which
 - whose
 - where
- The adverb "eventually" (line 6) means the same as:
 - ultimately
 - even so
 - after
 - never
- According to the second paragraph, the concern many users have is that:
 - Both standards are going to be used by everyone simultaneously.
 - Users are not going to have to choose with one to use.
 - Industries are going to adopt both standards.
 - Users are unsure about which is going to become an industry standard.
- What is the "dangerous business strategy" referred to in the end of the second paragraph?
 - Choosing one standard over another.
 - Trying to use both standards.
 - Waiting to see which approach will become an industry standard.
 - The fast change in the Internet of Things.
- Which of the words below is not a synonym of "hurdle" (line 9)?
 - difficulty
 - complication
 - ease
 - hindrance
- Which of the verbs below could replace the verb "permit" (line 11)?
 - halt
 - hinder
 - dissent

- (d) enable new changes.
10. The proposed solution in paragraph 3 is that:
- (a) Industries are going to have to choose their standards independently.
 - (b) Users are going to have to adapt to the
 - (c) Since both standards are incompatible, nothing can be done.
 - (d) It will be possible for both technologies to operate concomitantly.

Texto 2: Industry Adapting to Collaborative Robots

1 One of the first media properties I worked for back in the early 1990s was “Robotics World”. As a staff editor
2 on that publication at the time, I often heard how I had missed the glory days of the robotics industry back in
3 the early to mid 1980s when U.S. industrial adoption of robots was at its peak. By the early 1990s, most experts
4 thought U.S. industry was satiated with robots and no more large-scale implementations would be happening any
5 time soon.

6 Though things were quiet on the industrial robotics front for many years after that, robotics technology kept
7 advancing. Now, with the emergence _____ collaborative robots, we seem to be on the cusp of a new age of robotic
8 deployment that could well surpass the heady days of robot deployment in the 1980s.

9 To get more insight into this, I spoke with Chetan Kapoor, senior director of technology innovation at Yaskawa
10 Innovation Inc. Kapoor has led robotics research projects for DARPA and NASA and served as chief scientist at
11 the University of Texas at Austin. In his role at Yaskawa, he is focused _____ collaborative robots; development
12 of a common software platform to integrate robots, motion and drives; and creation of an interface that makes it
13 easier to program robots.

14 Kapoor will participate in the Collaborative Robotics panel at The Automation Conference and Expo 2016 to
15 be held May 24-25 in Chicago. For more information and to register, visit: www.theautomationconference.com.

As questões de 11 a 20 são referentes ao Texto 2.

11. The expression “at its peak” (line 3) means:
- (a) The lowest point of something.
 - (b) The highest point of something.
 - (c) A point of convergence.
 - (d) At the top of a mountain.
12. The expression “satiated with” (line 4) could be replaced, without changing the meaning of the text, by:
- (a) Saturated with.
 - (b) Disgusted with.
 - (c) Satisfied with.
 - (d) Left wanting with.
13. The word “Though” (line 6) is used to introduce an idea of:
- (a) Cause
 - (b) Consequence
 - (c) Addition
 - (d) Contrast
14. Which preposition completes the gap in “with the emergence _____ collaborative robots” (line 7)?
- (a) to
 - (b) about
 - (c) of
 - (d) on
15. The expression “on the cusp of” (line 7) means:
- (a) at the end of something.
 - (b) in the middle of something.
 - (c) between the end of one thing and the beginning of another.
 - (d) after something.
16. The verb “surpass” (line 8) means:
- (a) to be shorter than.
 - (b) to underachieve.
 - (c) to go beyond.
 - (d) to be superior.
17. What does the pronoun “this” (line 9) refer to?
- (a) The new age in robotics.
 - (b) The changes in the past.
 - (c) What happened in the 1980s.
 - (d) Robotic deployment.
18. Which preposition completes the gap in “he is focused _____ collaborative robots” (line 11)?
- (a) in
 - (b) at
 - (c) on
 - (d) with

19. In the first paragraph, the author's main point is that:

- (a) Nobody really cared much about robotics.
- (b) U.S. industries were looking for more ways of implements robots.
- (c) Many thought that by the 1990s the presence of robots in U.S. had reached its maximum.
- (d) There were no developments in sight.

20. The second and third paragraphs mention that:

- (a) Robotics is past its golden age.
- (b) Collaborative robotics is still a distant dream.
- (c) The introduction of collaborative robots will be a bigger change than ever.
- (d) There are changes to be made, but not in the foreseeable future.

PROVA DE CONHECIMENTOS ESPECÍFICOS SISTEMAS INTELIGENTES

21. As funções “AindaNemSei” e “cria” foram codificadas em Linguagem C e são apresentadas a seguir.

```
Lista* cria (int v) {
    Lista* p = (Lista*) malloc(sizeof(Lista));
    p->info = v;
    return p;
}

Lista* AindaNemSei (Lista* lst, int v){
    Lista* novo = cria(v);
    Lista* ant = NULL;
    Lista* p = lst;

    while (p != NULL && p->info > v) {
        ant = p;
        p = p->prox;
    }
    if (ant == NULL) {
        novo->prox = lst;
        lst = novo;
    } else {
        novo->prox = ant->prox;
        ant->prox = novo;
    }
    return lst;
}
```

Estas funções foram implementadas com base nas seguintes declarações de uma lista encadeada:

```
struct lista {
    int info;
    struct lista* prox;
};
typedef struct lista Lista;
```

Considere que já existe a funções de liberação de memória (a função Libera) e codificou-se a função principal abaixo:

```
int main(int argc, char *argv[]) {
    Lista* lis = NULL;
    lis = AindaNemSei (lis, 55);
    lis = AindaNemSei (lis, 10);
    lis = AindaNemSei (lis, 24);
    lis = AindaNemSei (lis, 13);
    libera(lis);
    return 0;
}
```

O resultado do programa é:

- (a) Uma lista encadeada simples com os elementos na seguinte ordem: 55, 10, 24 e 13.
- (b) Uma lista encadeada simples com os elementos na seguinte ordem: 10, 13, 24, e 55.
- (c) Uma lista encadeada simples com os elementos na seguinte ordem: 55, 24, 13, e 10.
- (d) Há erros de compilação, e portanto não há saída.

22. Analise o pseudo-código a seguir e responda à questão:


```
int func(int *x, int y, int z) {
    if (y > z) return 0;
    else {
        if (x[y] > 10) return (x[y] + func(x, y+1, z));
        else if (x[y] > 5) {
            x[y] += 10; return (func(x, y+1, z));
        } else return 0;
    }
}

int main(){
    int m[] = {1, 6, 12, 4, 11, 3, 4, 2, 14, 7, 10};
    int r = func(m, 2, 9);
}
```

Ao final da execução do programa, a sequência de elementos do vetor será:

- (a) [1,6,22,4,21,3,4,2,24,7,20]
 - (b) [1,16,22,4,11,3,4,2,14,7,10]
 - (c) [1,6,22,4,11,3,4,2,14,7,10]
 - (d) [1,6,12,4,11,3,4,2,14,7,10]
25. Tabelas de dispersão (tabelas hash) armazenam elementos com base no valor absoluto de suas chaves e em técnicas de tratamento de colisões. As funções de dispersão transformam chaves em endereços-base da tabela, ao passo que o tratamento de colisões resolve conflitos em casos em que mais de uma chave é mapeada para um mesmo endereço-base da tabela. Suponha que uma aplicação utilize uma tabela de dispersão com 23 endereços-base (índices de 0 a 22) e empregue $h(x) = x \bmod 23$ como função de dispersão, em que x representa a chave do elemento cujo endereço-base deseja-se computar. Inicialmente, essa tabela de dispersão encontra-se vazia. Em seguida, a aplicação solicita uma seqüência de inserções de elementos cujas chaves aparecem na seguinte ordem: 44, 46, 49, 70, 27, 71, 90, 97, 95.
- (a) Não há colisões diretas nessa tabela de dispersão.
 - (b) Apenas 49 e 95 tem colisão direta na posição 3
 - (c) Apenas 44 e 90 tem colisão direta na posição 21
 - (d) 49 e 95 tem colisão direta na posição 3, assim como 44 e 90 na posição 21.
26. Analise as afirmativas abaixo.
- I. A programação dinâmica é um método ascendente que aborda um dado problema subdividindo-o em problemas mínimos, soluciona esses subproblemas, guarda as soluções parciais, combina os subproblemas e sub-resultados para obter e resolver os problemas maiores, até recompôr e resolver o problema original.
 - II. A divisão e conquista é um método recursivo e, por isso, descendente que decompõe sucessivamente um problema em subproblemas independentes triviais, resolvendo-os e combinando as soluções em uma solução para o problema original.
 - III. Um algoritmo guloso sempre faz escolhas que parecem ser as melhores no momento, ou seja, escolhas ótimas locais acreditando que estas escolhas o levem a uma solução ótima global. Por essa estratégia, nem sempre asseguram-se soluções ótimas, mas, para muitos problemas, as soluções são ótimas. Os problemas ideais para essa estratégia não devem ter a propriedade de subestrutura ótima.
 - IV. A técnica de tentativa e erro (backtracking) efetua uma escolha ótima local, na esperança de obter uma solução ótima global.

A análise permite concluir que:

- (a) todas as afirmativas são verdadeiras.
- (b) todas as afirmativas são falsas.

- (c) apenas as afirmativas I e II são verdadeiras.
(d) apenas as afirmativas II e III são verdadeiras.
27. Os proprietários de um teatro necessitam de uma ferramenta de software para reserva de lugares. O desenvolvedor contratado verificou que as poltronas disponíveis para reserva são referenciadas pelo número da fila (a partir do nº 1) e pelo número da cadeira (a partir do nº 1) em cada fila, em uma representação matricial em que as linhas e colunas da matriz correspondem, respectivamente, às filas e às colunas de cadeiras. Embora o contexto seja o da organização matricial – N filas de cadeiras (linhas), cada uma contendo M cadeiras (colunas) –, a solução a ser implementada utilizará uma estrutura linear unidimensional (vetor), sendo, portanto, necessária uma conversão entre o lugar referenciado (número f da fila, número c da cadeira) e a posição real na estrutura de armazenamento (posição p no vetor).

Na situação apresentada, considere que a referida matriz seja armazenada no vetor segundo sua sequência de linhas, da primeira para a última, e, em cada linha, da primeira coluna para a última, e que a primeira posição no vetor tenha índice 0. Nessa situação, a posição p da poltrona do teatro localizada à fila de número f e à coluna de número c , é igual a:

- (a) $c + f \times M$
(b) $f + c \times M$
(c) $M \times (c - 1) + (f - 1)$
(d) $M \times (f - 1) + (c - 1)$
28. Observe a função recursiva a seguir, em linguagem C.

```
int Prova (int N){
    if (N == 0) return 0;
    else return (N*2-1 + Prova(N-1));
}
```

Considerando-se que essa função sempre será chamada com variável N contendo inteiros positivos, o seu valor de retorno será:

- (a) O fatorial do valor armazenado em N .
(b) O valor armazenado em N elevado ao quadrado.
(c) O somatório dos N primeiros números inteiros positivos.
(d) O somatório dos N primeiros números pares positivos.
29. Considere a seguinte imagem de 3 bits e seu respectivo histograma:

1	1	5	2	5	2	5	1	1	5
6	7	7	0	5	7	7	2	7	1
5	2	5	3	7	1	4	5	3	5
7	7	7	2	1	6	1	5	3	1
5	5	7	1	7	2	1	2	5	5
1	0	1	2	5	1	3	1	0	7
0	5	7	5	0	7	1	3	0	5
5	5	4	0	7	1	5	5	4	4
0	5	1	3	4	2	7	1	2	7
5	7	5	1	0	1	7	1	0	5

Histograma

Pixel	0	1	2	3	4	5	6	7
Número de ocorrência	10	22	10	6	5	26	2	19

Um dos algoritmos gulosos mais famosos da literatura é o Código de Huffman, usado para a compressão de dados. Descrito abaixo:

```

procedure ÁRVOREDEHUFFMAN(Lista de nós  $L$ ; inteiro  $n$ )
  for  $i = 1$  until  $n - 1$  do
    crie novo nó  $z$ 
    sejam  $x, y$  os dois primeiros nós em  $L$ 
     $f(z) = f(x) + f(y)$ 
    insira  $z$  em ordem em  $L$ 
    filho esquerdo de  $z =$  nó  $x$ 
    filho direito de  $z =$  nó  $y$ 
  end for
end procedure

```

OBS: Considere que a lista L é ordenada pela frequência em ordem não-decrescente. Seja $f(x)$ a frequência do nó x .

Usando a codificação de Huffman quantos bits são necessários para o armazenamento desta figura:

- (a) 300.
- (b) 272.
- (c) 150.
- (d) 172.

30. Julgue os itens a seguir, acerca dos algoritmos para ordenação:

- I. O algoritmo de ordenação por fusão (*MergeSort*) tem complexidade $O(n^2)$.
- II. Em um algoritmo de ordenação por inserção compara-se dois elementos adjacentes e troca-se suas posições, se o segundo elemento é menor do que o primeiro. Assim é feito até que não ocorra nenhuma troca.
- III. No Algoritmo Quicksort, a escolha do elemento pivô influencia o desempenho do algoritmo.
- IV. O Método da bolha e a ordenação por inserção tem a mesma complexidade computacional, $O(n^2)$.

Estão certos apenas os itens:

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

31. Considere a seguinte função:

```

procedure ORDENAÇÃOOPORFUSÃO( $A, p, r$ )
  if  $p < r$  then
     $q = \lfloor \frac{p+r}{2} \rfloor$ 
    ORDENAÇÃOOPORFUSÃO( $A, p, q$ )
    ORDENAÇÃOOPORFUSÃO( $A, q + 1, r$ )
    FUSÃODELISTA( $A, p, q, r$ )
  end if
end procedure

```

OBS: Considere que a fusão de duas listas (FusãodeLista) com um total de n elementos, pode ser feita com $n - 1$ operações.

Com relação à Ordenação por Fusão podemos afirmar que:

- (a) A função recursiva que descreve o número de operações do algoritmo é dada por $F(n) = 2F(n/2) + (n-1)$ e a ordem de complexidade é $O(n \log_2(n))$.
- (b) A função recursiva que descreve o número de operações do algoritmo é dada por $F(n) = 2F(n-1) + (n-1)$ e a ordem de complexidade é $O(n \log_2(n))$.

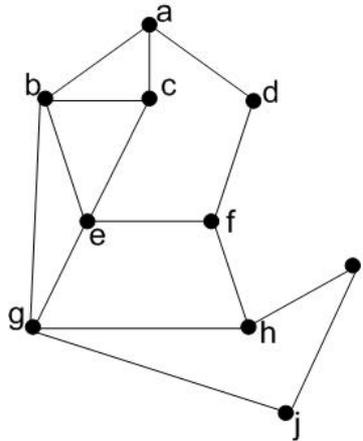
- (c) A função recursiva que descreve o número de operações do algoritmo é dada por $F(n) = 2F(n/2) + (n-1)$ e a ordem de complexidade é $O(n^2)$.
- (d) A função recursiva que descreve o número de operações do algoritmo é dada por $F(n) = 2F(n-1) + (n-1)$ e a ordem de complexidade é $O(n^2)$.

32. Julgue os itens a seguir, acerca de uma árvore binária:

- I. Considerando a raiz com nível 0, o número máximo de folhas no nível d é d^2 .
- II. O número máximo de nós na árvore é dado pela série $\sum_{i=0}^d 2^i$, onde d é o maior nível da árvore.
- III. A busca por um elemento em uma lista ordenada pode ser representada por uma árvore binária.
- IV. Em uma árvore binária cheia com n nós o comprimento da raiz à uma folha qualquer é dado por $\log_2(n+1) - 1$.

Estão certos apenas os itens

- (a) I e III.
- (b) III e IV.
- (c) I, III e IV.
- (d) II, III e IV.
33. Considerando busca em grafo, dê a ordem de inspeção dos nós através da busca em profundidade a partir do nó 'a' (seguindo a ordem lexicográfica), do grafo a seguir.



- (a) a, b, c, d, e, f, g, h, i, j.
- (b) a, b, c, e, f, d, h, i, j, g.
- (c) a, b, c, d, g, e, f, h, j, i.
- (d) a, b, g, j, i, h, f, e, c, d.
34. Seja G um Grafo direcionado e seja A sua matriz de adjacências. Podemos afirmar que o elemento da posição i, j na matriz A^2 mostra:
- (a) O número de caminhos com 2 saltos do nó i para o nó j .
- (b) Que na matriz de adjacências A existe ou não duas arestas entre i e j .
- (c) Que existem dois caminhos entre i e j .
- (d) Não tem significado físico para o grafo G .
35. O pseudo-código a seguir representa o Algoritmo de Dijkstra, onde: n é o número de nós do grafo G , S são os vértices para os quais se conhece o caminho mínimo e R são os vértices para os quais ainda não se conhece o caminho mínimo;

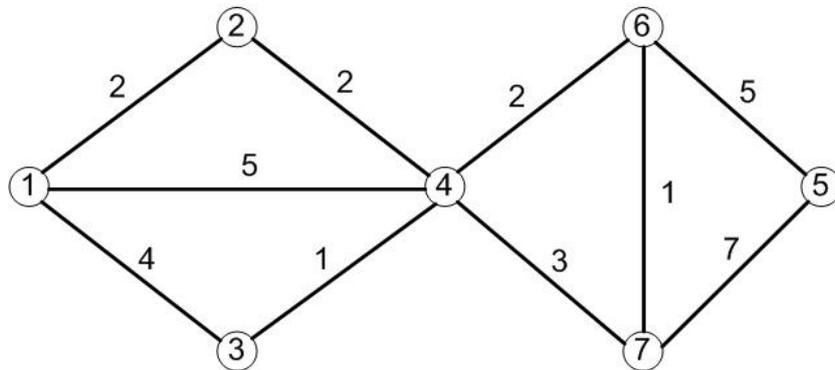
procedure DIJKSTRA($G = (N, A)$, c_{ij})

▷ c_{ij} é o custo entre os nós i e j

```

Iniciar variáveis:  $S \leftarrow \emptyset$ ;  $R \leftarrow N$ ;  $s \leftarrow 1$ ;  $d(i) \leftarrow \infty, \forall i \in N$ ;  $d(s) \leftarrow 0$ ;  $pred(s) \leftarrow 0$ ;
while  $|S| < n$  do
  Seja  $i$  tal que  $d(i) = \min\{d(j) : j \in R\}$ ;
   $S \leftarrow S \cup \{i\}$ ;
   $R \leftarrow R - \{i\}$ ;
  for  $j \in N$  com  $(i, j) \in A$  do
    if  $d(j) > d(i) + c_{ij}$  then
       $d(j) \leftarrow d(i) + c_{ij}$ ;
       $pred(j) \leftarrow i$ ;
    end if
  end for
end while
end procedure
  
```

Para o grafo abaixo diga o valor de cada variável do algoritmo após o término das iterações, iniciando no nó 1.



- (a) $d(1) = 0, d(2) = 2, d(3) = 4, d(4) = 4, d(5) = 11, d(6) = 6$ e $d(7) = 7$;
 $pred(1) = 0, pred(2) = 1, pred(3) = 1, pred(4) = 2, pred(5) = 6, pred(6) = 4, pred(7) = 4$.
- (b) $d(1) = 0, d(2) = 2, d(3) = 4, d(4) = 4, d(5) = 11, d(6) = 6$ e $d(7) = 7$;
 $pred(1) = 0, pred(2) = 1, pred(3) = 1, pred(4) = 2, pred(5) = 6, pred(6) = 4, pred(7) = 6$.
- (c) $d(1) = 0, d(2) = 2, d(3) = 4, d(4) = 4, d(5) = 11, d(6) = 6$ e $d(7) = 7$;
 $pred(1) = 0, pred(2) = 1, pred(3) = 1, pred(4) = 3, pred(5) = 6, pred(6) = 4, pred(7) = 4$.
- (d) $d(1) = 0, d(2) = 2, d(3) = 4, d(4) = 5, d(5) = 11, d(6) = 6$ e $d(7) = 7$;
 $pred(1) = 0, pred(2) = 1, pred(3) = 1, pred(4) = 3, pred(5) = 6, pred(6) = 4, pred(7) = 4$.
36. Seja A o conjunto de links apresentados pela busca da palavra “X” em um motor de buscas. Analogamente temos os conjuntos B e C dos links encontrados com a busca das palavras “Y” e “Z”, respectivamente. Se A, B e C são três conjuntos onde $n(A) = 25, n(B) = 18, n(C) = 27, n(A \cap B) = 9, n(B \cap C) = 10, n(A \cap C) = 6$ e $n(A \cap B \cap C) = 4$, (sendo $n(X)$ o número de elementos do conjunto X), determine o número de links encontrados pela busca (“X” ou “Y” e “Z”).
- (a) 11
 (b) 34
 (c) 12
 (d) 22
37. No jogo Daily3, são sorteados três números de 0 a 9 (permitindo repetições). O jogador marca três números em seu cartão de jogo e pode escolher como jogar, se em sequência (os números do jogador têm que ser iguais aos números sorteados na mesma ordem), ou em lote (os números do jogador têm que ser iguais aos números sorteados em qualquer ordem).
- I. a probabilidade de uma sequência é $1/1000$
 II. a probabilidade de um lote se forem sorteados 3 números distintos é $3!/1000$

III. a probabilidade de um lote se dois números forem iguais é $2!/1000$

Estão certos apenas os itens

- (a) I, II e III
- (b) I e II
- (c) I e III
- (d) II e III

38. Julgue os itens a seguir, acerca das Árvores de Decisão

- I. O Número de folhas de uma árvore de decisão para ordenar n elementos é $n!$
- II. A altura mínima de uma árvore de decisão para um algoritmo de ordenação com n elementos é $n!$
- III. A raiz de uma árvore de decisão para um algoritmo de busca binária agindo em uma lista ordenada com 11 itens representa a comparação do elemento desejado com o sexto item na lista.
- IV. Para um determinado conjunto de dados, a construção da árvore binária de busca é única.

Estão certos os itens:

- (a) I, II e III
- (b) I e II
- (c) I e III
- (d) II e III

39. A tabela abaixo representa uma árvore binária:

Nós	Filho esquerdo	Filho Direito
1	2	0
2	5	3
3	0	4
4	8	0
5	0	6
6	0	7
7	0	0
8	0	0

Indique a ordem de impressão dos seguintes algoritmos usados para percorrer árvores

- (a) Pré-Ordem: 7, 6, 5, 8, 4, 3, 2, 1; Ordem-Simétrica: 5, 6, 7, 2, 3, 8, 4, 1; Pós-Ordem: 1, 2, 5, 6, 7, 3, 4, 8.
- (b) Pré-Ordem: 1, 2, 5, 6, 7, 3, 4, 8; Ordem-Simétrica: 1, 2, 3, 4, 5, 6, 7, 8; Pós-Ordem: 7, 6, 5, 8, 4, 3, 2, 1.
- (c) Pré-Ordem: 1, 2, 4, 8, 5, 3, 6, 7; Ordem-Simétrica: 8, 4, 2, 5, 1, 6, 3, 7; Pós-Ordem: 8, 4, 5, 2, 6, 7, 3, 1.
- (d) Pré-Ordem: 1, 2, 5, 6, 7, 3, 4, 8; Ordem-Simétrica: 5, 6, 7, 2, 3, 8, 4, 1; Pós-Ordem: 7, 6, 5, 8, 4, 3, 2, 1.

Sejam: T uma árvore; T_i a i -ésima sub-árvore de T ; t o número de sub-árvores de T ; e r a raiz de T .

```

procedure ORDEM SIMÉTRICA(árvore  $T$ )
  ORDEM SIMÉTRICA( $T_1$ )
  escreva( $r$ )
  for  $i = 2$  until  $t$  do
    ORDEM SIMÉTRICA( $T_i$ )
  end for
end procedure

```

```

procedure PÓS ORDEM(árvore  $T$ )
  for  $i = 1$  until  $t$  do
    PÓS ORDEM( $T_i$ )
  end for

```

```

end for
escreva(r)
end procedure

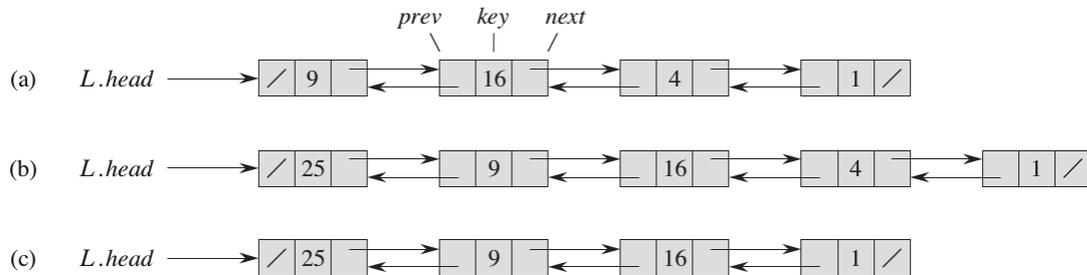
```

```

procedure PRÉORDEM(árvore T)
escreva(r)
for i = 1 until t do
PRÉORDEM(Ti)
end for
end procedure

```

40. Uma lista duplamente encadeada é uma estrutura de dados composta por dois tipos de objetos: (a) a cabeça da lista, e (b) as células que armazenam os dados da lista. A cabeça da lista pode estar vazia, i.e., contendo o valor NIL, o que indica que não há nenhum valor armazenado na lista, ou pode estar apontando para a primeira célula da lista. Cada célula da lista é composta por três elementos: *key*, que referencia o valor armazenado na célula; *prev*, que referencia a célula anterior – no caso da primeira célula, *prev* = NIL; e *next*, que referencia a próxima célula da lista – no caso da última célula, *next* = NIL. A figura abaixo ilustra uma lista duplamente encadeada que é submetida a uma operação de inserção e em seguida é realizada uma operação de deleção.



Na sub-figura (a) é mostrada a lista *L*, original com *L.head* representando a cabeça da lista *L*. Na sub-figura (b) podemos ver que foi inserida uma nova célula no início da lista contendo o valor 25. Na sub-figura (c) mostra-se a lista *L* depois que a célula contendo o elemento 4 foi removida.

A função `list_search`, listado abaixo, recebe uma chave *k* e retorna a célula da lista *L* que contém esta chave, caso a chave *k* não exista a função retorna NIL.

```

function LIST_SEARCH(L,k)
x ← L.head
while x ≠ NIL ∧ x.key ≠ k do
x ← x.next
end while
return x
end function

```

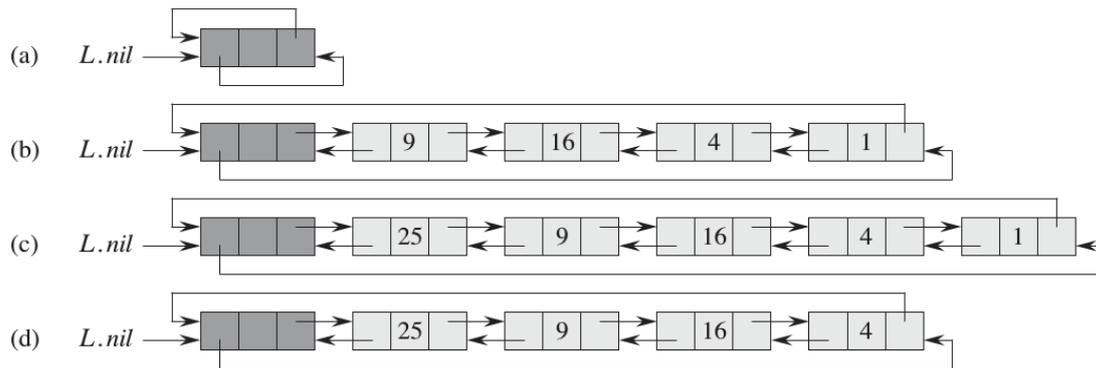
O procedimento `list_delete`, listada abaixo, recebe uma lista *L* e uma célula *x* de própria lista referenciada por *L*, e remove a célula *x* da lista *L*.

```

procedure LIST_DELETE(L,x)
if x.prev ≠ NIL then
x.prev.next ← x.next
else
L.head ← x.next
end if
if x.next ≠ NIL then
x.next.prev ← x.prev
end if
end procedure

```

Agora considere a variação da lista duplamente encadeada, chamada de **lista duplamente encadeada circular com sentinela** ilustrada na figura abaixo.



Nesta estrutura de dados, a primeira célula – o sentinela – não é usada para armazenar dados, ela é utilizada apenas para armazenar referências para a primeira e a última célula “reais” da lista. Nesta estrutura de dados a cabeça da lista não possui mais o campo *head*, ao invés disso, o primeiro elemento da lista é referenciado como $L.nil.next$ e o último elemento é referenciado como $L.nil.prev$. Assim, na figura acima, a sub-figura (a) representa uma lista vazia; a sub-figura (b) representa uma lista contendo os elementos 9, 16, 4 e 1; a sub-figura (c) representa a lista de (b) após a inserção do elemento 25 no início da lista; e a sub-figura (d) representa a lista em (c) após a remoção do elemento 1.

Considerando a lista duplamente encadeada circular com sentinela, conforme descrita acima, indique qual dos procedimentos abaixo melhor realiza a deleção da célula x passada como argumento.

- (a) **procedure** LIST_DELETE(L,x)
 if $x.prev \neq L.nil$ **then**
 $x.prev.next \leftarrow x.next$
 else
 $L.nil \leftarrow x.next$
 end if
 if $x.next \neq L.nil$ **then**
 $x.next.prev \leftarrow x.prev$
 end if
end procedure
- (b) **procedure** LIST_DELETE(L,x)
 $L.nil.prev \leftarrow x.prev$
 $x.prev.next \leftarrow x.next$
 $L.nil.next \leftarrow x.next$
 $x.next.prev \leftarrow x.prev$
end procedure
- (c) **procedure** LIST_DELETE(L,x)
 if $x.next = L.nil$ **then**
 $L.nil.prev \leftarrow x.prev$
 end if
 $x.prev.next \leftarrow x.next$
 if $x.prev = L.nil$ **then**
 $L.nil.next \leftarrow x.next$
 end if
 $x.next.prev \leftarrow x.prev$
end procedure
- (d) **procedure** LIST_DELETE(L,x)
 if $x.prev \neq L.nil$ **then**
 $x.prev.next \leftarrow x.next$
 end if
 if $x.next \neq L.nil$ **then**
 $x.next.prev \leftarrow x.prev$
 end if
end procedure