

Nome:
RG:

QUESTÕES

Questão 1. (peso 1) Quantas senhas de no mínimo 5 caracteres e no máximo 7 caracteres podem ser construídas quando é permitido usar as 5 vogais minúsculas do alfabeto e 10 algarismos, sendo que o primeiro caractere da senha é, obrigatoriamente, uma vogal e que podemos repetir caracteres? Explique sua resposta.

Questão 2. (peso 0,75) Nesse quadrimestre, uma certa disciplina da graduação em Ciência da Computação da UFABC precisou ofertar seis turmas devido à demanda reprimida. Os professores X , Y e Z estão responsáveis por duas turmas cada um, sendo uma turma de manhã e outra de noite. As quantidades de alunos inscritos em cada turma são apresentadas na tabela a seguir.

	Professor X	Professor Y	Professor Z
Manhã	45	5	32
Noite	67	2	4

Dentre esses estudantes matriculados nessa disciplina, escolhe-se um ao acaso. Qual é a probabilidade desse estudante estar matriculado na turma noturna, dado que ele escolhe aulas apenas com o professor X ? Explique sua resposta.

Questão 3. (peso 1) Explique:

(a) Por que $\log_{2b} y = \frac{\log_b y}{1 + \log_b 2}$.

(b) Por que $\frac{4 + \log_2 x}{3} = \log_2 \sqrt[3]{16x}$ para todo $x > 0$.

Questão 4. (peso 1) Seja x uma variável que indica a quantidade de produtos produzidos por dia em uma determinada fábrica. Seja $c(x) = \frac{1}{4}x^2 + 35x + 25$ uma função que determina o custo de produção, em reais, e $v(x) = 50 - \frac{1}{2}x$ uma função que determina o preço de venda, em reais, dos produtos. Apresente a quantidade total de produtos que a fábrica deve produzir em um dia para maximizar o lucro total. Justifique sua resposta.

Questão 5. (peso 2) Descreva em sua linguagem de programação estruturada favorita uma estrutura de árvore binária. Em seguida, assumindo que a árvore é de busca, apresente funções para:

1. Devolver a quantidade de nós armazenados.
2. Devolver a altura da árvore.
3. Devolver a quantidade de nós folha.
4. Devolver o menor elemento.
5. Devolver o maior elemento.

Para cada função, explique, de forma sucinta, qual é a ideia do que está sendo feito.

Questão 6. (peso 1) Nessa questão, considere que se uma sequência (a_1, \dots, a_n) representa uma fila, então o elemento a_1 está na frente da fila e, de forma geral, a_i está há mais tempo na fila do que a_{i+1} , para $1 \leq i < n$. Se uma sequência (a_1, \dots, a_n) representa uma pilha, então o elemento a_1 é seu topo e, de forma geral, a_i está há menos tempo na pilha do que a_{i+1} , para $1 \leq i < n$.

Uma empresa desenvolvedora de softwares para biologia computacional trabalha com sequenciamento de DNA, sendo que cada sequência contém, portanto, as letras A, C, T, G . Em um determinado trabalho, a sequência inicial (A, G, T, C, A, G, T, T) ficou armazenada na estrutura de dados X . Cada elemento foi retirado de X e inserido na estrutura Y , sendo que ao final a sequência armazenada em Y foi (T, T, G, A, C, T, G, A) . Por fim, cada elemento foi retirado de Y e inserido em Z , sendo que a sequência final ficou (T, T, G, A, C, T, G, A) .

Responda:

1. Dada uma sequência $A = (a_1, \dots, a_n)$, mostre a sequência resultante ao inserir em A o elemento x quando A é uma pilha e quando A é uma fila.
2. Dada uma sequência $A = (a_1, \dots, a_n)$, mostre a sequência resultante ao remover um elemento de A quando A é uma pilha e quando A é uma fila.
3. Sabendo que $X \in \{\text{pilha, fila}\}$, $Y \in \{\text{pilha, fila}\}$ e $Z \in \{\text{pilha, fila}\}$, determine X, Y e Z que sejam condizentes com o cenário descrito na empresa. Explique sua resposta.

Questão 7. (peso 1) Uma lista duplamente ligada possui a seguinte definição de nó:

```
1 typedef node {
2     int informacao;
3     node pointer prox, ant;
4 };
```

Como a função a seguir deve ser completada para inverter uma lista duplamente ligada?

```

1  function inverte(node pointer h) {
2      node pointer p, q;
3      if h != NULL {
4          p = h->prox;
5          h->prox = NULL;
6          h->ant = p;
7          while p != NULL {
8              ???
9          }
10     }
11 }

```

Questão 8. (peso 1,5) Realize um teste de mesa, mostrando o passo a passo da execução do código a seguir.

```

1  function f(int n, int pointer k) {
2      int p, q;
3      if n < 2 {
4          *k := 0;
5      } else {
6          n := f(n-1, &p) + f(n-2, &q);
7          *k := p + q + 1;
8      }
9      write(n, *k);
10     return n;
11 }
12
13 function main() {
14     global int n := 4;
15     write(f(3, &n), n);
16 }

```

Questão 9. (peso 0,75) Assinale a afirmativa incorreta:

1. Seja $A[1..n]$ um vetor não ordenado de inteiros com um número constante k de valores distintos. Então existe algoritmo de ordenação por contagem que ordena A em tempo linear.
2. Seja $A[1..n]$ um vetor não ordenado de inteiros com um número constante k de valores distintos. Então o limite inferior para um algoritmo de ordenação por comparações para

ordenar A é de $O(n \log n)$.

3. Seja $A[1..n]$ um vetor não ordenado de inteiros, sendo que cada inteiro tem no máximo d dígitos e cada dígito assume um valor dentre um número constante k de valores distintos. Então o problema de ordenar A tem limite inferior $O(n)$.
4. Seja $A[1..n]$ um vetor não ordenado de inteiros, sendo que cada inteiro tem no máximo d dígitos e cada dígito assume um valor dentre $O(n)$ valores distintos. Então o problema de ordenar A tem limite inferior $O(n \log n)$.
5. Seja $A[1..n]$ um vetor não ordenado de inteiros com um número constante k de valores distintos. Então um algoritmo de ordenação por comparações ótimo para ordenar A tem complexidade $O(n \log n)$.