



# PROCESSO SELETIVO VAGAS RESIDUAIS 2015

## UFBA



**06**

**ESTRUTURA DE DADOS**

**CIRCUITOS DIGITAIS E ARQUITETURA  
DE COMPUTADORES**

**REDAÇÃO**

---

## INSTRUÇÕES

Para a realização das provas, você recebeu este Caderno de Questões, uma Folha de Respostas para as Provas I e II e uma Folha de Resposta destinada à Redação.

### 1. Caderno de Questões

- Verifique se este Caderno de Questões contém as seguintes provas:  
Prova I: ESTRUTURA DE DADOS — Questões de 01 a 35  
Prova II: CIRCUITOS DIGITAIS E ARQUITETURA DE COMPUTADORES — Questões de 36 a 70  
Prova de REDAÇÃO
- Qualquer irregularidade constatada neste Caderno de Questões deve ser imediatamente comunicada ao fiscal de sala.
- Nas Provas I e II, você encontra apenas um tipo de questão: objetiva de proposição simples. Identifique a resposta correta, marcando na coluna correspondente da Folha de Respostas:

V, se a proposição é verdadeira;

F, se a proposição é falsa.

**ATENÇÃO:** Antes de fazer a marcação, avalie cuidadosamente sua resposta.

**LEMBRE-SE:**

- A resposta correta vale 1 (um), isto é, você **ganha** 1 (um) ponto.
- A resposta errada vale -0,5 (menos meio ponto), isto é, você **não ganha** o ponto e ainda **tem descontada**, em outra questão que você acertou, essa fração do ponto.
- A ausência de marcação e a marcação dupla ou inadequada valem 0 (zero). Você **não ganha nem perde** nada.

### 2. Folha de Respostas

- A Folha de Respostas das Provas I e II e a Folha de Resposta da Redação são pré-identificadas. Confira os dados registrados nos cabeçalhos e assine-os com caneta esferográfica de **TINTA PRETA**, sem ultrapassar o espaço próprio.
- **NÃO AMASSE, NÃO DOBRE, NÃO SUJE, NÃO RASURE** ESSAS FOLHAS DE RESPOSTAS.
- Na Folha de Respostas destinada às Provas I e II, a marcação da resposta deve ser feita preenchendo-se o espaço correspondente com caneta esferográfica de **TINTA PRETA**. Não ultrapasse o espaço reservado para esse fim.

Exemplo de Marcação  
na folha de Respostas

01	<input type="checkbox"/>	F
02	<input checked="" type="checkbox"/>	V
03	<input checked="" type="checkbox"/>	V
04	<input type="checkbox"/>	F
05	<input checked="" type="checkbox"/>	V

- O tempo disponível para a realização das provas e o preenchimento das Folhas de Respostas é de 4 (quatro) horas e 30 (trinta) minutos.
-

---

ESTAS PROVAS DEVEM SER RESPONDIDAS PELOS CANDIDATOS AO SEGUINTE CURSO:

- SISTEMAS DE INFORMAÇÃO

---

## PROVA I — ESTRUTURA DE DADOS

### QUESTÕES de 01 a 35

#### INSTRUÇÃO:

Para cada questão, de **01** a **35**, marque na coluna correspondente da Folha de Respostas:

**V**, se a proposição é verdadeira;

**F**, se a proposição é falsa.

A resposta correta vale 1 (um ponto); a resposta errada vale  $-0,5$  (*menos* meio ponto); a ausência de marcação e a marcação dupla ou inadequada valem 0 (zero).

#### Questão 01

Um algoritmo com complexidade assintótica  $O(n^7)$  é sempre mais eficiente que outro com complexidade assintótica  $O(2^n)$ .

#### Questão 02

Em análise de complexidade de algoritmos, é comum o estudo de três casos:

- melhor caso, menor tempo de execução sobre todas as entradas de tamanho  $n$ ;
- pior caso, maior tempo de execução sobre todas as entradas de tamanho  $n$ ;
- caso médio, média dos tempos de execução de todas as entradas de tamanho  $n$ .

#### Questão 03

Supondo-se que um algoritmo X possui função de custo  $f(n) = 81n^3$ , um outro algoritmo Y possui função de custo  $f(n) = n^7$ , pode-se afirmar que o algoritmo X é sempre mais custoso que o algoritmo Y, qualquer que seja o valor de  $n$ .

### QUESTÕES de 04 a 07

Para responder a essas questões, considere a estrutura de dados definida a seguir, em linguagem C.

```
struct tNo { int dado;
            struct tNo* prox;
            };
typedef struct tNo No;

struct tListaEncadeada { No* inicio;
                        int tamanho;
                        };
typedef struct tListaEncadeada ListaEncadeada;
```

---

## Questão 04

A função **F1**, tal como está codificada a seguir, implementa uma operação de **busca** de um valor inteiro **n** na **ListaEncadeada L**.

```
No* F1(ListaEncadeada L, int n) {  
  
    if (L.inicio == NULL)  
        return NULL;  
  
    No* posAtual = L.inicio;  
  
    while ( (posAtual != NULL) && (posAtual->dado != n) )  
        posAtual = posAtual->prox;  
  
    return posAtual;  
  
}
```

## Questão 05

A função **F2**, descrita a seguir, implementa uma operação de **remoção** de um valor inteiro **n** na **ListaEncadeada L**.

```
bool F2(ListaEncadeada* L, int n) {  
  
    No* pos = L -> inicio;  
  
    if ( pos == NULL )  
        return false;  
  
    while ((pos) && (pos -> dado != n))  
        pos = pos -> prox;  
    if ( pos == L->inicio ) {  
        L->inicio = L->inicio->prox;  
    }  
    else {  
        No* posAtual = L->inicio;  
        No* posAnterior = NULL;  
  
        while ( posAtual != pos) {  
            posAnterior = posAtual;  
            posAtual = posAtual->prox;  
        }  
        posAtual->prox = posAnterior->prox;  
    }  
  
    free (pos);  
    L->tamanho--;  
    return true;  
  
}
```

## Questão 06

A função **F3**, tal como está codificada a seguir, é capaz de **imprimir** os valores do campo **dado** de todos os nós de uma **ListaEncadeada**, na ordem em que estão armazenados na lista, caso seja chamada, passando-se como parâmetro um ponteiro para o primeiro nó da lista.

```
void F3(No *no) {  
  
    if ( no == NULL)  
        return;  
    else {  
        printf("%d\n", no->dado);  
        F3(no->prox);  
    }  
  
}
```

---

## Questão 07

Seja **L** uma **ListaEncadeada**, cujos nós possuem os valores do campo **dado** em ordem crescente.

A função **F4**, como codificada a seguir, é capaz de **inserir** um **novoNo**, com valor do campo **dado** igual a **n**, e manter a ordem da lista.

Considere que a função **alocaNovoNo** recebe um valor inteiro **n** como parâmetro e tenta alocar uma área para armazenar uma variável do tipo **No**. Caso a alocação seja feita com sucesso, o endereço do novo **No** é retornado com os campos **dado** e **prox** preenchidos com os valores **n** e **NULL**, respectivamente. Caso contrário o endereço **NULL** é retornado.

```
bool F4(ListaEncadeada* L, int n) {
    No *novoNo, *atual;
    if (! (novoNo=alocaNovoNo( n)))
        return false;
    atual = L->inicio;
    while ( atual->prox != NULL )
        atual = atual->prox;
    atual->prox = novoNo;
    L->tamanho++;
    return true;
}
```

## Questão 08

O tipo abstrato de dados **Pilha** pode ser implementado de forma eficiente utilizando-se uma estrutura de dados formada por um vetor e um campo inteiro que indica o índice do elemento do topo.

## Questão 09

O tipo abstrato de dados **Fila** pode ser implementado de forma eficiente utilizando uma estrutura de dados formado por um vetor e dois inteiros, que indicam, respectivamente, os índices dos elementos do início e do fim da fila.

## Questão 10

**Pilhas** e **Filas** são estruturas de dados lineares que se diferenciam basicamente pela sua política de inserção e de remoção de elementos, **FIFO** e **LIFO**, respectivamente.

## Questão 11

Em uma árvore, a raiz possui nível 1 e, cada nó possui, como o seu nível, 1 a mais que o nível de seu pai, portanto a altura dessa árvore pode ser definida como o maior valor de um nível de um nó dessa árvore.

## Questão 12

Os percursos em pré-ordem, pós-ordem e ordem simétrica são diferentes formas de se percorrer uma árvore binária. Esses percursos se diferenciam no momento em que um nó é visitado durante o percurso.

## QUESTÕES 13 e 14

Para responder a essas questões considere um **Heap** implementado a partir de um estrutura de dados do tipo vetor, onde o nó-raiz é armazenado no índice 0.

## Questão 13

A relação entre o índice **i** de um nó e os índices de seus dois filhos, esquerdo e direito, é dada, respectivamente, pelas expressões  $2i + 1$  e  $2i + 2$ .

## Questão 14

A relação entre o índice **i** de um nó e o índice de seu pai é dada pela expressão  $\left\lfloor \frac{(i-1)}{2} \right\rfloor$ .

---

### Questão 15

O **algoritmo de ordenação por seleção** possui complexidade  $O(n)$  para melhor caso, que ocorre quando o conjunto já se encontra ordenado.

### Questão 16

O **algoritmo de ordenação por inserção** possui complexidade  $O(n)$  para melhor caso, que ocorre quando o conjunto já se encontra ordenado.

### Questão 17

O **algoritmo de ordenação QuickSort** possui complexidade  $O(n \cdot \log_2(n))$  para melhor caso, que acontece quando o conjunto já se encontra ordenado.

### Questão 18

O **algoritmo de ordenação por seleção** possui complexidade  $O(n^2)$  para pior caso, que ocorre quando o conjunto se encontra ordenado, porém na ordem inversa da desejada.

### Questão 19

O **algoritmo de ordenação por inserção** possui complexidade  $O(n^2)$  para pior caso, que ocorre quando o conjunto se encontra ordenado, porém na ordem inversa da desejada.

### Questão 20

O **algoritmo de ordenação QuickSort** possui complexidade  $O(n \cdot \log_2(n))$  para pior caso, que ocorre quando o conjunto se encontra ordenado.

### Questão 21

O **algoritmo de ordenação QuickSort** garante seu melhor funcionamento quando o processo de pivoteamento é feito com base na mediana do conjunto a ser ordenado.

### Questão 22

Os **algoritmos de ordenação externa** procuram minimizar a transferência de dados entre a memória principal e a memória secundária, utilizando estruturas em árvore para representar a localização dos dados em disco.

### Questão 23

O **algoritmo de Intercalação Balanceada de Vários Caminhos** se divide em duas etapas básicas: primeiro a criação de blocos ordenados nas fitas disponíveis; em seguida, a aplicação de sucessivas intercalações dos blocos até que exista apenas um bloco.

### Questão 24

O número de passadas que o **algoritmo de Ordenação por Intercalação Balanceada de f-Caminhos** faz é dado pela expressão  $P(N) = \left\lceil \log_r \left( \frac{N}{m} \right) \right\rceil$ , em que **N** é o número de registros do arquivo original e **m**, o número de palavras que um registro ocupa em memória interna.

### Questão 25

O **algoritmo de busca sequencial** possui complexidade de melhor caso  $O(1)$  e pior caso  $O(n)$ .

### Questão 26

O **algoritmo de busca binária** possui complexidade de melhor caso  $O(1)$ , que corresponde a encontrar a chave procurada na primeira comparação.

### Questão 27

O **algoritmo de busca binária** possui complexidade de pior caso  $O(\log_2(n))$ , que corresponde à busca em um conjunto de chaves não ordenado.

---

## QUESTÕES de 28 a 30

Para responder a essas questões, considere:

- a definição da estrutura de dados que representa uma **ArvoreBinariaBusca**.

```
struct tNode { int data;
               struct tNode* left;
               struct tNode* right;
             };

typedef struct tNode TreeNode;

TreeNode* ArvoreBinariaBusca;
```

- r a raiz de uma subárvore de uma **ArvoreBinariaBusca**.

### Questão 28

A função **F1** realiza a remoção de todos os nós de uma subárvore **r**.

```
void F1( TreeNode** r) {

    if ( *r != NULL) {
        free(*r);
        F1(&(*r) ->left);
        F1(&(*r) ->right);
    }
}
```

### Questão 29

A função **F2** realiza a busca pelo valor **n** por todos os nós.

```
TreeNode* F2( TreeNode* r, int n) {

    if ( r == NULL )
        return r;

    if ( r->data == n )
        return r;

    if ( r->data > n )
        return F2(r->left, n);
    else
        return F2(r->right, n);
}
```

### Questão 30

Considere que o parâmetro **r** da função **F3** contém o endereço da raiz de uma **ArvoreBinariaBusca**. Adicionalmente, considere que a função **createNode** aloca uma área de memória para um **TreeNode**, atribui os valores **n**, **NULL** e **NULL** aos campos **dado**, **left** e **right**, respectivamente, e retorna ao endereço alocado.

A função **F3** descrita promove a inserção de um valor **n** em uma **ArvoreBinariaBusca**.

```
void F3(TreeNode** r, int n) {

    if ( (*r) == NULL ) {
        (*r) = createNode (n);
        return;
    }
    if ( (*r) ->data > n )
        if ((*r)->left != NULL)
            F3 (&(*r) ->left, n);
        else
            (*r) ->left = createNode (n);
    else
        if ((*r) ->right != NULL)
            F3 (&(*r)->right, n);
        else
            (*r) ->right = createNode (n);
}
```

---

### Questão 31

Uma **Árvore AVL** é dita balanceada quando, para cada nó da árvore, a diferença entre as alturas das suas subárvores (direita e esquerda) não é maior do que dois.

### Questão 32

O procedimento de inserção de um novo nó em uma **Árvore AVL Balanceada** é feito em dois passos: primeiro, o algoritmo de inserção de um novo nó em uma árvore binária de busca é aplicado para determinar onde o novo nó será inserido. A seguir, uma rotação simples ou uma rotação dupla sempre deve ser aplicada, para que a árvore retorne à sua condição de balanceamento.

### Questão 33

A operação de rotação dupla de uma **Árvore AVL Balanceada** é uma operação custosa, uma vez que envolve mudanças em todos os descendentes e ascendentes do nó que causou o desbalanceamento.

### Questão 34

Dos algoritmos de busca em memória interna, os algoritmos do tipo **Hash** são os mais eficientes, pois permitem sempre obter o registro desejado em tempo  $O(1)$ .

### Questão 35

Em uma **Árvore B** todos os nós folhas possuem o mesmo nível.

---

## PROVA II — CIRCUITOS DIGITAIS E ARQUITETURA DE COMPUTADORES

### QUESTÕES de 36 a 70

#### INSTRUÇÃO:

Para cada questão, de 36 a 70, marque na coluna correspondente da Folha de Respostas:

**V**, se a proposição é verdadeira;

**F**, se a proposição é falsa.

A resposta correta vale 1 (um ponto); a resposta errada vale -0,5 (*menos* meio ponto); a ausência de marcação e a marcação dupla ou inadequada valem 0 (zero).

#### Questão 36

O número decimal 4048, ou seja  $(4048)_{10}$ , convertido para a base binária, com 16 bits de representação, é igual a  $(0000111111010000)_2$ .

#### Questão 37

O número binário de 16 bits  $(11101111000100011010)_2$  convertido para a base hexadecimal é igual a  $(EF11A)_{16}$ .

#### Questão 38

Considerando a expressão  $X = (1000011011)_2 - (0001101101)_2$ , que representa uma subtração binária, o valor final de X é igual a  $(111001000)_2$ .

#### Questão 39

Considerando a expressão  $X = (CAFE)_{16} + (1F)_{16} - (A2)_{16}$ , em base hexadecimal, o valor final de X é igual a  $(CA7D)_{16}$ .

#### Questão 40

Se a representação em complemento de 2, com 8 bits, de um número decimal X é 11010001, então esse número decimal X é igual a -80.

#### Questão 41

O número decimal -71, representado com 8 bits nas formas sinal-magnitude, complemento de 1 (C1) e complemento de 2 (C2), é, respectivamente, igual a 11000111, 10111000 e 10111001.

---

## RASCUNHO

### Questão 42

No padrão IEEE 754 de representação de números binários, em ponto flutuante, com 32 bits, o número decimal -99.625 é representado no seguinte formato:

Bit de sinal	Expoente polarizado	Mantissa normalizada
1	1000101	1000111010000000000000

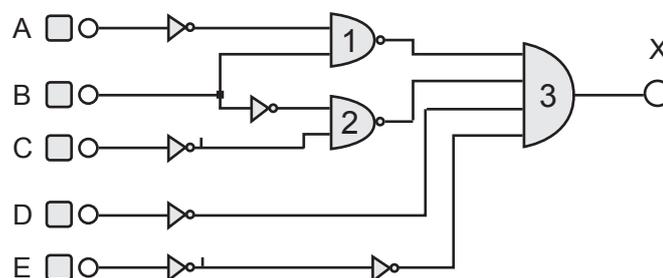
### Questão 43

O número binário 1001011001 corresponde ao número 1101110101 em código Gray.

### Questão 44

O código BCD 1000100100110101 corresponde ao número 7931 em base decimal.

### QUESTÕES 45 e 46



### Questão 45

Considere o circuito formado por duas portas NAND (1 e 2) e uma porta AND (3). Com os valores das entradas A=0, B=1, C=1, D=0 e E=1, o valor produzido na saída X é igual ao valor da entrada B.

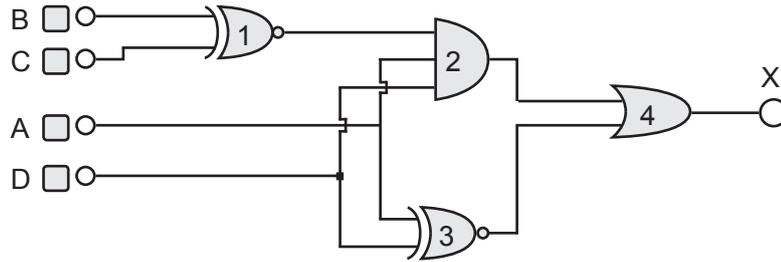
### Questão 46

A expressão algébrica não simplificada para a saída X do circuito é  $X = \neg(A \cdot B) \cdot \neg(B \cdot C) \cdot D \cdot E$ , considerando-se o sinal de exclamação (!) que precede uma variável ou termo a operação NOT.

---

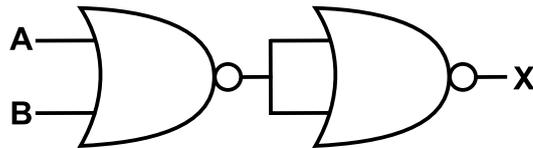
## RASCUNHO

### Questão 47



Submetendo-se as entradas  $A=1$ ,  $B=1$ ,  $C=0$  e  $D=0$ , ao circuito formado por duas portas XNOR (1 e 3), uma porta AND (2) e uma porta OR (4), o valor apresentado na saída X é igual ao valor da entrada C.

### Questão 48



Considerando a universalidade da porta NOR, o circuito apresentado é semanticamente equivalente (ou seja, implementa a mesma função) a uma porta AND.

### Questão 49

O teorema  $A + AB = A$  corresponde a uma das leis de absorção da Álgebra Booleana.

### Questão 50

Aplicando-se exclusivamente os Teoremas de DeMorgan à expressão  $!((X.Y+W).(X.W.Y))$ , obtém-se a expressão  $!(X.Y) + !W$ , considerando o sinal de exclamação (!) como sendo a operação NOT.

## RASCUNHO

---

## QUESTÕES 51 e 52

Para responder a essas questões, considere a Tabela-verdade a seguir.

Entradas				Saída
A	B	C	D	X
0	0	0	0	0
0	0	0	1	1
0	0	1	0	0
0	0	1	1	0
0	1	0	0	1
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	0
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

### Questão 51

A expressão de produto de somas é composta pelos maxtermos  $M_0$ ,  $M_2$ ,  $M_3$  e  $M_8$ .

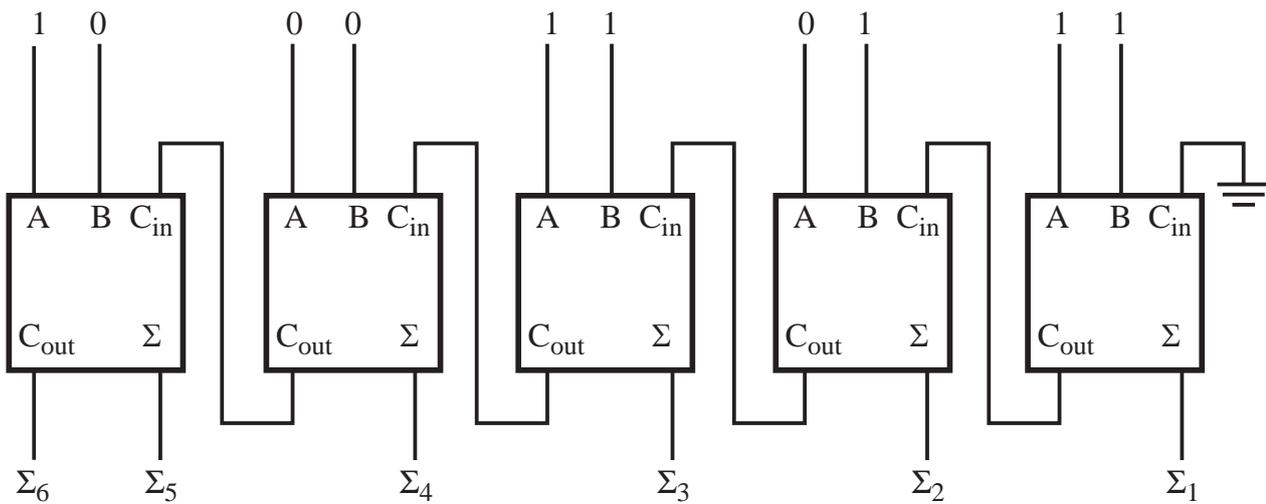
### Questão 52

Um mapa de Karnaugh-Veitch para as saídas em nível ALTO gera a expressão algébrica  $(!A.!C.!D) + (A.!C.D) + (A.!B.C) + B$ , considerando o sinal de exclamação (!) como sendo a operação NOT.

---

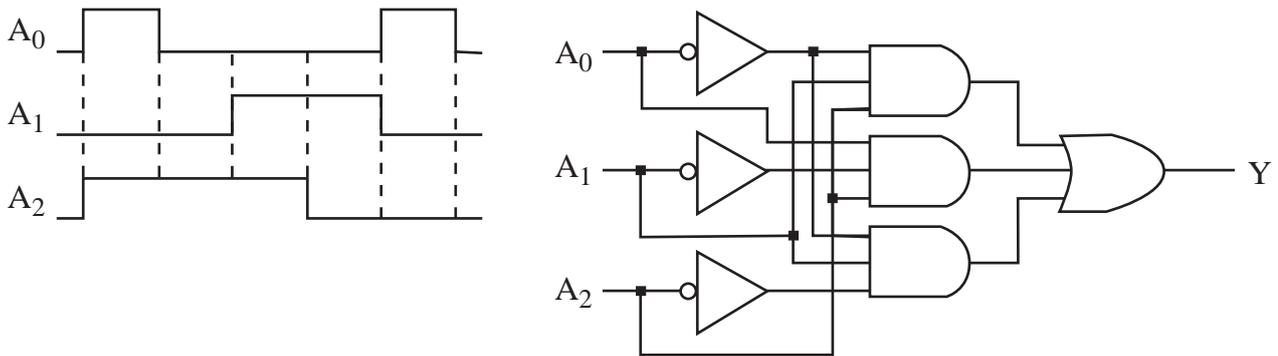
## RASCUNHO

### Questão 53



Na operação do somador completo, considerando as entradas ilustradas na figura, a saída (soma completa) do circuito para as saídas  $\Sigma_1 \Sigma_2 \Sigma_3 \Sigma_4 \Sigma_5 \Sigma_6$  é, na ordem, 110001.

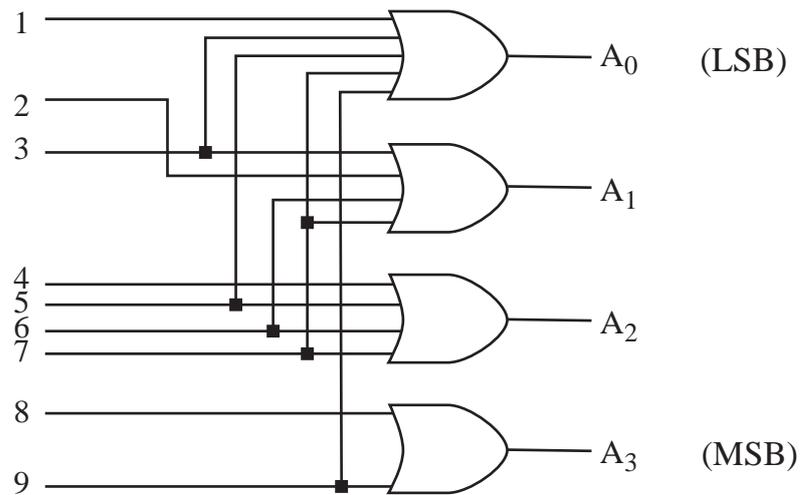
### Questão 54



Nas formas de onda da figura,  $T_0$  é a linha preta tracejada mais à direita e  $T_5$  é a linha preta tracejada mais à esquerda. No instante  $T_0$ , as entradas  $A_0 A_1 A_2$  são todas iguais a zero. No instante  $T_5$ , as entradas  $A_0 A_1 A_2$  valem, respectivamente, 101. Quando aplicadas à lógica de decodificação da figura, as entradas produzem as saídas 0 e 1, respectivamente, para os instantes  $T_0$  e  $T_5$ .

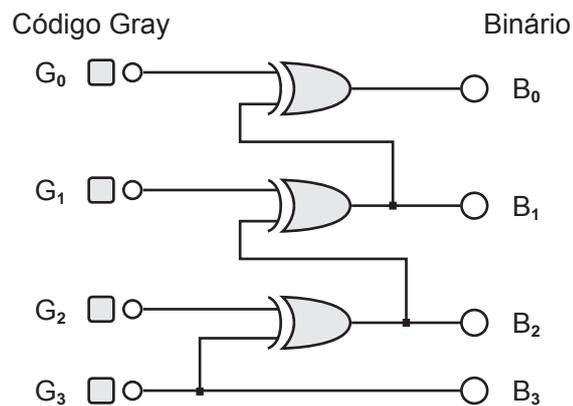
### RASCUNHO

### Questão 55



No codificador de decimal para BCD mostrado na figura, quando as entradas 3 e 9 estão em nível ALTO e as demais entradas estão em nível BAIXO, o circuito produz uma saída BCD válida.

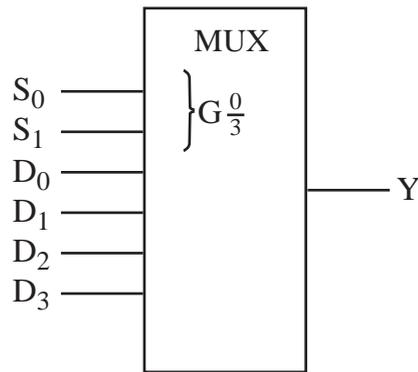
### Questão 56



Na figura apresentada, se o código Gray 1001 for aplicado às entradas  $G_3G_2G_1G_0$ , respectivamente, o valor gerado nas saídas binárias  $B_3B_2B_1B_0$  será, respectivamente, 0110.

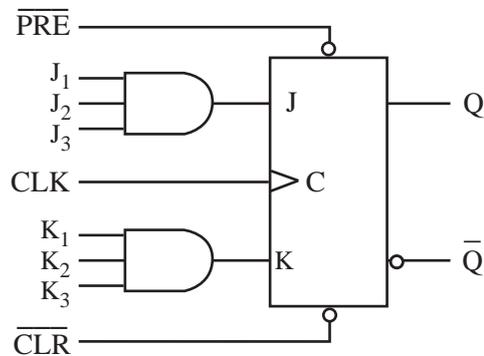
**RASCUNHO**

### Questão 57



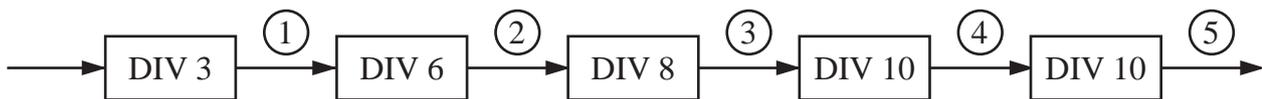
No multiplexador da figura, a saída  $Y$  é igual ao valor da entrada  $D_3$  quando os valores das entradas  $D_0 D_1 D_2 D_3 S_0 S_1$  forem, respectivamente, 011010.

### Questão 58



A saída  $Q$  do flip-flop JK da figura é 0 no estado inicial. Ao serem aplicadas as entradas  $J_1 J_2 J_3$  e  $K_1 K_2 K_3$ , todas em nível ALTO, a saída  $\overline{Q}$  passa a ter o valor zero, considerando o sinal de exclamação (!) como sendo a operação NOT.

### Questão 59



O módulo total da configuração dos contadores da figura é igual a 37.

## RASCUNHO

---

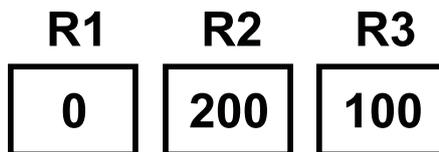
### Questão 60

O conceito de programa armazenado, no qual as instruções de um programa e os dados a serem manipulados por esse programa ficam armazenados simultaneamente em memória, é uma das características introduzidas pela arquitetura de computador empregada na máquina de von Neumann.

### Questão 61

Quando um processador acessa a memória cache em busca de um determinado dado e o encontra, levando este dado para um de seus registradores, considera-se que houve um *cache miss*.

### QUESTÕES 62 e 63



Registrador base



Registrador de índice



Memória	
Endereço	Conteúdo
100	300
110	20
120	1
200	200
1100	55
1110	15
1200	110
1300	45
1400	50

---

## RASCUNHO

---

### Questão 62

Considerando uma arquitetura que utilize o modo de endereçamento indireto por registrador e o formato de instrução <OPCODE, OP<sub>1</sub>, OP<sub>2</sub>, OP<sub>3</sub>>, sendo OPCODE o código (mnemônico) da instrução, OP<sub>1</sub> o operando destino e OP<sub>2</sub> e OP<sub>3</sub> os operandos-fonte, então a instrução <ADD R<sub>1</sub>, R<sub>2</sub>, R<sub>3</sub>> atribui ao registrador R<sub>1</sub> o valor 300.

### Questão 63

Considerando uma arquitetura que utilize o modo de endereçamento “base com índice e deslocamento” e o formato de instrução <OPCODE, OP<sub>1</sub>, OP<sub>2</sub>, OP<sub>3</sub>>, sendo OPCODE o código (mnemônico) da instrução, OP<sub>1</sub> o operando destino e OP<sub>2</sub>, OP<sub>3</sub> os operandos-fonte, então a instrução <SUB R<sub>1</sub>, R<sub>2</sub>, R<sub>3</sub>> atribui ao registrador R<sub>1</sub> o valor 95.

### Questão 64

O mecanismo de interrupções empregado nas arquiteturas de computadores pode utilizar diferentes tipos de interrupções, classificadas como interrupções de programa, de temporização (*timer*), de entrada e saída e de falhas de hardware.

### Questão 65

O emprego de controle microprogramado e de múltiplos modos de endereçamento são características marcantes das arquiteturas CISC (*complex instruction-set computers*).

### Questão 66

Um *pipelining* RISC básico é composto por cinco estágios: busca da instrução (IF), decodificação da instrução (ID), execução da instrução ou cálculo de operandos (EX), acesso a operandos na memória (MEM) e escrita de resultados (WB).

### Questão 67

Número da instrução	Código da instrução (OPCODE)	Operando 1	Operando 2	Operando 3	Descrição
I1	LW	\$t0	0(\$t0)		// \$t0 = memória[0 + \$t0]
I2	LW	\$t1	20(\$t1)		// \$t1 = memória[20 + \$t1]
I3	ADD	\$s3	\$t0	\$t1	// \$s3 = \$t0 + \$t1
I4	SW	\$s3	12(\$t1)		// Memória[12 + \$t1] = \$s3
I5	SUB	\$s3	\$s4	\$t1	// \$s3 = \$s4 - \$t1

LW = load word, ADD = soma, SUB = subtração, SW = store word

O trecho de código apresentado possui uma dependência de dados verdadeira (ou de leitura após escrita - RAW) entre as instruções I3 e I4 no acesso ao registrador \$t1.

---

## RASCUNHO

---

**Questão 68**

Uma unidade de processamento gráfico (GPU) é caracterizada por centenas de núcleos de processamento (cores) capazes de processar uma grande quantidade de fluxos de execução (*threads*) de forma simultânea.

**Questão 69**

Uma implementação superescalar capaz de executar duas instâncias de cada estágio do seu *pipelining* em paralelo é dita de grau 2.

**Questão 70**

Se a execução de um programa, de forma sequencial, consome 4 minutos e 32 segundos num determinado processador, e a versão paralela desse programa requer 2 minutos e 17 segundos, logo o fator de aceleração (*speed up*) para esse programa é de 1.99.

---

**RASCUNHO**

---

## PROVA DE REDAÇÃO

### INSTRUÇÕES:

- Escreva sua Redação com caneta de tinta AZUL ou PRETA, de forma clara e legível.
- Caso utilize letra de imprensa, destaque as iniciais maiúsculas.
- O rascunho deve ser feito no local apropriado do Caderno de Questões.
- Na Folha de Resposta, utilize apenas o espaço a ela destinado.
- Será atribuída a pontuação ZERO à Redação que
  - se afastar do tema proposto;
  - for apresentada em forma de verso;
  - for assinada fora do local apropriado;
  - apresentar qualquer sinal que, de alguma forma, possibilite a identificação do candidato;
  - for escrita a lápis, em parte ou na sua totalidade;
  - apresentar texto incompreensível ou letra ilegível.

Os textos a seguir devem servir como ponto de partida para a sua Redação.

#### I.

Uma nação se faz com pessoas imbuídas de propósitos comuns, voltadas para o progresso social, com pleno exercício da justiça e da igualdade de direitos e de deveres de seus cidadãos. A consciência coletiva de uma sociedade resulta dos valores de uma nação. O que se passa na consciência como pensamento coletivo das pessoas é reflexo de sua cultura, da opinião pública e

5 – do estágio de desenvolvimento da sociedade.

Essa consciência coletiva pode colocar uma sociedade em letargia ou, de outro modo, fazê-la progredir. Linchamentos, queima de ônibus, manifestações com quebra-quebra, depredação de patrimônio público, bem como tudo que implique um modo coletivo de ser, seja com resultados positivos ou não para a sociedade, são resultantes da consciência coletiva. O que circula nas

10 – redes sociais, o que aparece nas diversas mídias, o que pensa a classe dominante, o que dizem os artistas mais consagrados e o que falam os formadores de opinião, o que emana das sub-regiões urbanas no formato de opinião comum são também indícios do conteúdo da consciência coletiva.

[...] Mais do que qualquer outro vetor, a opinião pública, pelo seu poder de penetração e por sua linguagem que traduz o que se passa na consciência coletiva, pode contribuir em muito para a

15 – educação visando à plenitude da cidadania.

NOVAES, A. Consciente coletivo. **A Tarde**, Salvador, 20 maio 2015. Caderno Opinião, p. A3.

#### II.

O consumo declinante de produtos culturais entre brasileiros, identificado em pesquisa recente divulgada pela Federação do Comércio (Fecomércio) do Rio de Janeiro, reflete não apenas a atual crise econômica, mas um nó crítico do sistema educacional.

O estudo aponta, entre outros, um dado estarrecedor: 70% dos brasileiros não leram um

5 – livro sequer em 2014. O uso da internet, amplificado pelos smartphones, é apontado como um dos responsáveis pela queda na leitura, principalmente entre os jovens. Nos países desenvolvidos, 13 é o número médio de livros lidos anualmente por habitante, enquanto, no Brasil, são dois.

Alguns acreditam que isso pode ser explicado pelo fato de, ao iniciar tardiamente o seu processo de escolarização, nos anos 1960, o país ter saltado do analfabetismo para o audiovisual,

10 – sem conseguir formar uma cultura de leitura.

A pesquisa indica também que o volume de frequentadores de cinema diminuiu, embora as idas ao teatro tenham dobrado em relação a 2009. Apesar disso, 89% não assistiram a nenhuma peça entre 2013 e 2014. [...]

Uma nação que não consome cultura tem dificuldade de entender e discutir em

15 – profundidade questões que dizem respeito a todos. Ao se distanciar dos livros, o Brasil se torna um país raso.

AZIZ, B. Exclusão cultural. **A Tarde**, Salvador, 17 maio 2015. Caderno Opinião, p. A3. Editorial.

#### PROPOSTA

A partir da leitura dos fragmentos acima, produza um **texto argumentativo** em que você estabeleça **uma relação entre cultura, educação e direitos humanos**, ressaltando as ações que o homem brasileiro pode promover no sentido de formar uma nação pautada pela ética e pela cidadania.

---

## RASCUNHO







**UNIVERSIDADE FEDERAL DA BAHIA**  
PROGRAD/COORDENAÇÃO DE SELEÇÃO E ORIENTAÇÃO  
Rua Dr. Augusto Viana, 33 – Canela  
Cep. 40110-060 – Salvador/BA  
Telefax (71) 3283-7820 – E-mail: [ssoa@ufba.br](mailto:ssoa@ufba.br)  
Site: [www.vagasresiduais.ufba.br](http://www.vagasresiduais.ufba.br)