



INSTRUÇÕES GERAIS AOS CANDIDATOS

- O tempo total para realização das provas é de **1 hora e 30 minutos**.
- Ao término da prova, o candidato deverá devolver o cartão resposta.
- É imprescindível verificar no cartão resposta o número de inscrição do candidato no espaço reservado para tal.

A IDENTIFICAÇÃO DOS CANDIDATOS EM TODAS AS PÁGINAS DEVERÁ SER FEITA **APENAS** PELO NÚMERO DE INSCRIÇÃO.

- As respostas deverão ser transpostas para o cartão resposta com caneta de tinta azul ou preta. Não serão consideradas as respostas que não estiverem transcritas no cartão resposta, bem como não serão consideradas respostas rasuradas.
- A Prova de Base Computacional é constituída por 10 questões objetivas¹.
- Cada questão objetiva tem somente uma resposta correta.
- A prova deve ser feita sem consulta e sem empréstimo de material.
- Verifique se sua prova contém 10 questões, assim como o cartão de respostas.
- **Não** é permitido o uso de calculadora, celular ou qualquer outro aparelho durante a realização da prova. É vedado o empréstimo de qualquer material entre os candidatos.

Boa Prova !

¹As questões desta prova foram adaptadas ou extraídas de edições anteriores do POSCOMP.



CARTÃO DE RESPOSTAS

INSCRIÇÃO N^o: ____

Questão	Alternativa			
1	A	B	C	D
2	A	B	C	D
3	A	B	C	D
4	A	B	C	D
5	A	B	C	D
6	A	B	C	D
7	A	B	C	D
8	A	B	C	D
9	A	B	C	D
10	A	B	C	D



1. A saída do trecho de código abaixo é.

```
Algoritmo
  declare  $i, j, k, cont$  : inteiros;
   $i \leftarrow 0$  ;       $j \leftarrow -9$ ;       $k \leftarrow 7$ ;
  Para  $cont \leftarrow 4$  até  $cont < 11$  faça{
     $i \leftarrow cont + 5$ ;
    enquanto ( $k > 0$ ){
       $k \leftarrow k - 1$ ;
       $j \leftarrow k + j/2$ ;
    fim enquanto
  fim para
  Escreva "i:", i, "j:", j, "k:", k;
fim algoritmo
```

- A. i:1j:4k:1
- B. i:15j:1k:0
- C. i:10j:14k:2
- D. i:13j:2k:3

2. Analise o seguinte programa descrito na forma de pseudocódigo:

```
Algoritmo
  declare  $X[10]$ ,  $n$ ,  $i$ ,  $aux$ ,  $flag$ : numérico;
  para  $i \leftarrow 1$  até 10 faça
    leia  $X[i]$ ;
   $n \leftarrow 1$ ;
   $flag \leftarrow 1$ ;
  enquanto ( $n \leq 10$  E  $flag = 1$ ) faça
  inicio
     $flag \leftarrow 0$ ;
    para  $i \leftarrow 1$  até 9 faça
    inicio
      se ( $X[i] < X[i + 1]$ ) então
      inicio
```



```
    flag ← 1;  
    aux ← X[i];  
    X[i] ← X[i + 1];  
    X[i + 1] ← aux;  
fim-se  
fim-para  
n ← n + 1;  
fim-enquanto  
para i ← 1 até 10 faça  
    escreva X[i];  
fim-algoritmo
```

Esse programa realiza a ordenação decrescente de um vetor de números inteiros, que implementa o algoritmo de ordenação:

- A. rápida
- B. por troca
- C. por seleção
- D. por inserção

3. Considere o algoritmo a seguir.

```
Algoritmo  
declare valor, i, x, D, N, j, termo: numérico;  
valor ← 1;  
i ← 2;  
x ← 2;  
repita  
    N ← xi;  
    j ← 1;  
    D ← 2;  
repita  
    D ← D * j;  
    j ← j + 1;  
se j >= i então
```



```
        interrompa;  
    fim se  
fim repita  
termo ←  $(-1)^{(i+1)} * N/D$ ;  
valor ← valor + termo;  
i ← i + 1;  
se (i > 5) então  
    interrompa;  
fim se  
fim repita  
escreva “Valor =”, valor;  
Fim Algoritmo.
```

Assinale a alternativa que apresenta, corretamente, o conteúdo da variável “valor” ao final da execução do algoritmo.

- A. 0,2220
- B. 0,3330
- C. 1,2220
- D. 1,3330

4. Considere o trecho de algoritmo, apresentado na forma de uma pseudolinguagem (Português Estruturado), a seguir. Assuma que no comando “leia A, B, C, D;” os valores lidos são, respectivamente, 12, 25, 96 e 15 e a função RESTO (x,y) apresenta o resto da divisão inteira de x por y

Algoritmo

```
leia A, B, C, D;  
resp ← 1;  
i ← 2;  
repita  
    se (RESTO (A,i)=0 ou RESTO(B,i)=0 ou RESTO(C,i)=0 ou RESTO(D,i)=0)  
    então  
        resp ← resp * i;  
        se (RESTO (A,i)=0) então  
            A ← A/i;
```



```
    fim se
      se (RESTO (B,i)=0) então
         $B \leftarrow B/i;$ 
      fim se
      se (RESTO (C,i)=0) então
         $C \leftarrow C/i;$ 
      fim se
      se (RESTO (D,i)=0) então
         $D \leftarrow D/i;$ 
      fim se
    senão
       $i \leftarrow i + 1;$ 
    fim se
    se (A=1 e B=1 e C=1 e D=1) então
      interrompa;
    fim se
  fim repita
  escreva "Resposta =", resp;
fim algoritmo
```

Assinale a alternativa que apresenta, corretamente, o conteúdo da variável resp, impresso no comando “escreva “Resposta =”, resp;”.

- A. 480
- B. 800
- C. 1200
- D. 2400

5. A técnica computacional que demanda a forma de tratamento de colisões é conhecida como:

- A. Algoritmo merge-sort
- B. Algoritmo guloso
- C. Tabela Hash
- D. Árvore binária de busca



6. Um estudante de computação precisa resolver um problema bastante importante, que é executar as operações que estão descritas abaixo, cuja estrutura é uma pilha. Tão logo ele retire algum elemento desta pilha, estes deverão ser inseridos em uma fila, cuja entrada é pela esquerda e a saída, pela direita. Assinale a alternativa que contém a sequência correta de entrada dos elementos na fila.

PUSH P
PUSH E
PUSH R
PUSH T
PUSH O
POP
POP
PUSH S
PUSH O
PUSH L
POP
POP
POP

- A. S - O - L - T - O
- B. O - T - R - E - P
- C. P - E - R - T - O
- D. O - T - L - O - S

7. A função abaixo computa a soma dos n primeiros números inteiros não negativos:

```
funcao soma(n: inteiro): inteiro{  
  se  $n = 0$  então  $soma \leftarrow 0$   
  senão -----  
}
```

A parte que falta para completar a condição senão é:

- A. enquanto $(n! = 0)\{soma \leftarrow soma + soma(n + 1)\}$
- B. $soma \leftarrow n + soma(n)$



- C. $soma \leftarrow (n - 1) + soma(n - 1)$
- D. $soma \leftarrow n + soma(n - 1)$

8. Professor Bezerra propôs o seguinte algoritmo de ordenação, chamado de Super Merge, similar ao merge-sort: divide o vetor em 4 partes do mesmo tamanho (ao invés de 2, como é feito no merge-sort). Ordene recursivamente cada uma das partes e depois intercale-as por um procedimento semelhante ao procedimento de intercalação do merge-sort. Qual das alternativas abaixo é verdadeira?

- A. Super Merge não está correto. Não é possível ordenar quebrando o vetor em 4 partes.
- B. Super Merge está correto, mas consome tempo da mesma ordem que o merge-sort ($O(\text{merge-sort})$).
- C. Super Merge está correto, mas consome tempo de ordem maior que o merge-sort (maior que $O(\text{merge-sort})$).
- D. Super Merge está correto, mas consome tempo de ordem menor que o merge-sort (menor que $O(\text{merge-sort})$).

9. As estruturas de dados lineares (fila, pilha e lista) são muito utilizadas para resolver problemas computacionais. Cada uma dessas estruturas pode ser implementada com diferentes características e atendem a diferentes tipos de problemas. Sobre as características dessas estruturas de dados, análise as afirmativas a seguir.

- (I) Em uma pilha, o último elemento a entrar é o primeiro a sair.
- (II) Em uma fila, o primeiro elemento a entrar é o último a sair.
- (III) Uma lista permite que as inserções possam ser feitas em qualquer lugar (posição), mas as remoções, não.
- (IV) Em uma lista circular com encadeamento simples, o primeiro elemento aponta para o segundo e para o último.
- (V) Para remover um elemento de uma lista duplamente encadeada, deve-se alterar o encadeamento dos elementos anterior e próximo ao elemento removido.

São corretas as afirmativas:

- A. I e IV
- B. III e IV
- C. I e V
- D. III e V

10. Os algoritmos a seguir representam os três caminhamentos para árvores binárias.



Centro Federal de Educação Tecnológica Celso Suckow da Fonseca - CEFET/RJ
Programa de Pós-Graduação em Ciência da Computação
PROCESSO SELETIVO 2019.3
PROVA DE BASE COMPUTACIONAL

`caminhamento(binário)`

`se binário.esquerda ≠ NULL então caminhamento(binário.esquerda)`

`escrever binário.valor`

`se binário.direita ≠ NULL então caminhamento(binário.direita)`

`caminhamento(binário)`

`escrever binário.dado`

`se binário.esquerda ≠ NULL então caminhamento(binário.esquerda)`

`se binário.direita ≠ NULL então caminhamento(binário.direita)`

`caminhamento(binário)`

`se binário.esquerda ≠ NULL então caminhamento(binário.esquerda)`

`se binário.direita ≠ NULL então caminhamento(binário.direita)`

`escrever binário.valor`

Assinale a alternativa que contém os nomes dos 3 caminhamentos, respectivamente.

- A. pré-ordem, pós-ordem, em-ordem
- B. pré-ordem, em-ordem, pós-ordem
- C. pós-ordem, pré-ordem, em-ordem
- D. em-ordem, pré-ordem, pós-ordem