

### Questão 1

Explique o que é análise estática e análise dinâmica. Dê exemplos de como revisões e testes se complementam.

### Questão 2

Dê 2 exemplos de como revisões e testes se complementam. (1 pt)

### Questão 3

Considere o trecho de código abaixo que contém anomalias e viola certos padrões de programação.

a) realize uma inspeção do código abaixo e explique 3 destes problemas.

b) uma inspeção de código rigorosa elimina a necessidade de testes? Dê 2 justificativas para sua resposta.

```
int TOTAL_CONVERSOES;

long Conversao (short baseCalculo, int opcao) {
    long Resultado;

    if (opcao == 1 || opcao == 2 || opcao == 3) {
        Resultado = *(( long *) &baseCalculo );
    } else {
        Resultado = *(( int *) &baseCalculo );
    }

    TOTAL_CONVERSOES += 1;
    return Resultado;
}
```

### Questão 4

Quais as diferenças e similaridades entre as seguintes técnicas de garantia de qualidade: teste, medição e inspeção de código fonte?

### Questão 5

Quais as diferenças e similaridades entre as seguintes técnicas de garantia de qualidade: teste, medição e inspeção de código fonte?

### Questão 6

Explique os passos para que seja avaliado a cobertura de teste em um código em C usando a ferramenta **gcov**. Para se avaliar a qualidade dos testes de um programa, basta-se medir o grau de cobertura usando ferramentas como o **gcov**? Justifique com base no conceito e técnicas de teste de caixa fechada e de caixa aberta.

### Questão 7

O trecho de código abaixo apresenta um esquema de programa de pesquisa em tabela. Cada tabela é percorrida por um iterador. O acesso aos elementos da tabela se dá através da função ObterValor que depende do estado corrente do iterador:

```
tpRefElem ProcurarElem( tpTabela Tabela,
                        tpValor ValorProcurado )
{
    //Esquema de algoritmo para pesquisa em tabela
    tpRefElem Achou;
```

```

tpRefElem Corrente;

Achou    = ELEM_TABELA_NIL;
Corrente = DefinirPrimeiro( Tabela, Valor Procurado );
while ( Existe( Tabela, Corrente ) )
{
    if ( ObterValor( Tabela, Corrente) == ValorProcurado )
    {
        Copiar( Achou, Corrente );
        break;
    } else {
        Corrente = DefinirProximo( Tabela, Corrente,
ValorProcurado );
    } //if
} //while

```

1. Defina a(s) assertiva(s) de entrada e assertiva(s) de saída da função. (1.5 ponto)
2. Decomponha o trecho em blocos, defina o arrasto e calcule a expressão regular dos caminhos a serem testados. (1.5 ponto)
3. Segundo o critério de cobertura de decisões, defina quais são as decisões a serem examinadas conforme os casos de indução da(s) repetição(ões). (0.5 ponto)
4. Com base nos caminhos definidos a partir da expressão definida no item b, defina o caso mínimo de testes semânticos que sejam completos. (1.5 ponto)

### Questão 8

Supondo que você tenha uma empresa que fabrica software em larga escala. Um dos diferenciais do seu produto é que ele é tido como um dos mais confiáveis do mercado. E quando acontecem problemas, sua equipe consegue detectar rapidamente a solução, pois vocês disponibilizam uma versão de produção e outra de desenvolvimento através da técnica de **instrumentação**. Explique utilizando exemplo prático (e.g. seu recém lançado Jogo) os passos e diretivas que sua empresa usa para **implementação e compilação** dessa técnica em seus programas.

### Questão 9

Considere o algoritmo do MERGE abaixo: o parâmetro  $A$  é um vetor de inteiros e  $p$ ,  $q$  e  $r$  são índices de enumeração dos elementos desse vetor tal que  $p \leq q < r$ . Assuma que os subarranjos  $A[p..q]$  e  $A[q+1..r]$  estão em seqüência ordenada crescente. Assumindo que a entrada  $A$  é definida e  $p$ ,  $q$  e  $r$  respeitam a equação descrita anteriormente:

1. Faça assertivas de saída com relação ao vetor para os valores  $L[i]$  e  $R[j]$  ao final da execução algoritmo.
2. Defina e justifique o arrasto para o loop entre as linhas 12 a 17, calcule a expressão regular geradora de caminhos desse trecho.
3. Defina o caso mínimo de testes semânticos da solução em (b) para que sua análise seja completa.
4. Uma vez implementado o algoritmo abaixo, ele estaria sujeito a algum dos padrões de falta? Se sim, qual (is)?

```

MERGE( $A, P, q, r$ )
1  $n_1 \leftarrow q - p + 1$ 
2  $n_2 \leftarrow r - q$ 
3 criar arranjos  $L[1..n_1 + 1]$  e  $R[1..n_2 + 1]$ 
4 for  $i \leftarrow 1$  to  $n_1$ 
5     do  $L[i] \leftarrow A[p + i - 1]$ 
6 for  $j \leftarrow 1$  to  $n_2$ 
7     do  $R[j] \leftarrow A[q + j]$ 
8  $L[n_1 + 1] \leftarrow \infty$ 
9  $R[n_2 + 1] \leftarrow \infty$ 
10  $i \leftarrow 1$ 
11  $j \leftarrow 1$ 
12 for  $k \leftarrow p$  to  $r$ 
13     do if  $L[i] \leq R[j]$ 
14         then  $A[k] \leftarrow L[i]$ 
15              $i \leftarrow i + 1$ 
16     else  $A[k] \leftarrow R[j]$ 
17          $j \leftarrow j + 1$ 

```

#### Questão 10

Explique sucintamente os passos para ser gerada uma documentação utilizando o Doxygen.