

PROGRAMAÇÃO SISTEMÁTICA – TURMA A

LISTA II

Questão 1

Para cada afirmacao abaixo, indique se e verdadeira ou falsa, e justifique com argumentos sua resposta.

- a) “Um programador implementou inicialmente uma funcao X sem tratar eventuais excecoes que podem ocorrer no contexto desta funcao. Porem, em uma segunda etapa, o programador, ciente da falta de robustez no codigo original da funcao X, decide incluir tratadores de excecoes no codigo da mesma. Esta nova versao da funcao e chamada de X'. De forma a satisfazer um padrao de completeza de testes por arestas, pode se concluir que a versao X' precisa de menos casos de teste do que a versao original X da funcao.”
- b) “Uma das principais vantagens do uso de um esquema de algoritmo e que, uma vez que a argumentacao da corretude do mesmo ja foi realizada, nao e necessario realizar testes do codigo que implementa tal esquema.”
- c) Uma empresa desenvolveu um programa que deve ser amplamente robusto. Os seguintes instrumentos devem ser utilizados durante o desenvolvimento: (i) assertivas executaveis com pre- e pos-condicoes associadas com os parametros de cada funcao, (ii) contadores de passagem por funcao. A empresa vendera tal programa de duas formas – com e sem instrumentacao habilitada no programa entregue – de forma a atender requisitos de diferentes clientes. A conclusao e que tal empresa deve usar adaptadores (wrappers) para cada modulo instrumentado pois e a forma mais modular de permitir a escolha flexivel de inclusao (ou nao) dos instrumentos acima no momento da compilacao do programa final a ser entregue ao cliente.”

Questao 2

Examine o codigo da funcao comparar() abaixo, podendo assumir que esta sintaticamente correto. O codigo realiza uma analise comparativa de 3 valores positivos com o objetivo de encontrar um unico maior dentre os três numeros. Responda as questoes, apresentadas a seguir, relacionadas a satisfacao de criterios de seleção de casos de teste.

- a) Produza o fluxograma que represente os possiveis fluxos de execucao da funcao. As entradas dos cinco casos de teste abaixo representam um conjunto suficiente e minimo para satisfazer o criterio de cobertura de instrucoes? Justifique.

num1 = 3, num2 = 2, num3 = 2

num1 = 3, num2 = 4, num3 = 3

num1 = 3, num2 = 4, num3 = 5

num1 = 4, num2 = 5, num3 = 5

num1 = 5, num2 = 5, num3 = 5

- b) Os três caminhos (arcos de execucao) abaixo descrevem casos de teste abstratos para a funcao comparar(). Note que os rótulos (letras) estão associados com instrucoes e/ou blocos de codigo da funcao. Complete o conjunto de caminhos abaixo de forma a obter um conjunto minimo de casos de teste para satisfazer o padrao de cobertura de arestas. Qual o numero minimo de casos de teste que foi necessario para satisfazer tal padrao?

ABC, ABDEFGI, ABDEJK

- c) Apresente a *expressao regular* geradora de caminhos que permita identificar os casos de teste necessários para satisfacao do criterio de cobertura de caminhos. Use os rotulos (letras) associados

acima ao código da função `comparar()`. Qual seria o caminho mais longo e o mais curto que poderiam ser gerados a partir desta expressão?

1. #include <stdio.h>	
2.	
3. int comparar() {	
4. int num1, num2, num3, maior = 0;	A
5. printf ("Entre 3 numeros positivos");	
6. scanf ("%d %d %d", &num1, &num2, &num3);	
7. if ((num1 > num2) && (num1 > num3)) {	B
8. maior = num1;	C
9. return maior;	
10. } else {	D
11. if (num2 == num3) {	E
12. printf ("Nao existe maior ");	F
G 13. if (num3 == num1) { printf ("... e todos sao iguais!"); }	H
15. return maior;	I
16. }	
17. if ((num2 > num3) && (num2 != num1)) {	J
18. maior = num2;	K
19. return maior;	
20. }	
21. if ((num3 > num2) && (num3 != num1)) {	L
22. maior = num3;	M
23. return maior;	
24. }	
25. printf ("Nao existe maior ");	N
26. return maior;	
25. }	
26. }	

Questao 3

Considere uma estrutura de lista ligada circular, generica e unicamente encadeada. Apresente o modelo fisico e assertivas estruturais para tal estrutura. Em seguida, crie uma nova versao de tal modelo fisico de forma a tornar a estrutura auto-verificavel. (lembrete: uma lista circular e um tipo de lista encadeada na qual o ultimo elemento tem como proximo o primeiro elemento da lista, formando um circulo)

Questao 4

Usuarios da funcao de ordenacao abaixo decidiram realizar a argumentacao da corretude. A funcao implementa uma das formas alternativas de ordenacao baseada no metodo da bolha. A funcao recebe como parametros o numero de elementos e o ponteiro do primeiro elemento do vetor que se deseja ordenar. Para evitar que o processo de trocas continue mesmo depois de o vetor estar ordenado, a funcao interrompe tal processo quando houver uma passagem inteira sem trocas. Como voce foi o projetista original da funcao, pediram que voce determine todas as assertivas necessarias e suficientes para a argumentacao desta funcao. Apresente tambem quais seriam as proposicoes basicas para tais usuarios realizarem a argumentacao. Importante: voce nao precisa apresentar a descricao da argumentacao, nem apontar uma falta no codigo caso existir.

```

/* Ordenação bolha (2a. versão) */
void bolha2 (int n, int* v)
{
    int i, j;
    for (i=n-1; i>0; i--) {
        int troca = 0;
        for (j=0; j<i; j++)
            if (v[j]>v[j+1]) { /* troca */
                int temp = v[j];
                v[j] = v[j+1];
                v[j+1] = temp;
                troca = 1;
            }
        if (troca == 0) /* nao houve troca */
            return;
    }
}

```

Questao 5

Examine o código a seguir da função que realiza múltiplas análises de dois inteiros. Produza o fluxograma (baseado em vértices e arestas), apresentado em sala de aula, e descreva um conjunto mínimo de casos de teste que você geraria ao utilizar um critério de seleção de casos de teste baseado em cobertura de arestas para a função abaixo.

```

int analisar(int a, int b)
{
    if( a > b ) {
        printf("%d eh maior que %d\n", a, b);
        if ( a > ( b + 5 ))
            printf("%d eh muito maior que %d\n", a, b);
        a = a - 6;
    } else {
        printf("%d eh menor que %d\n", a, b);
        if ( a < ( b - 5 ))
            printf("%d eh muito menor que %d\n", a, b);
        b = b - 6;
    }

    if(a > b)
        printf("%d ainda eh maior que %d\n", a, b);
    else
        printf("%d ainda eh menor que %d\n", a, b);

    return 0;
}

```

Questao 6

Considere o esquema de algoritmo de pesquisa binária discutido em sala de aula e fornecido abaixo.

a) Suponha que você deve selecionar casos de teste (usando o padrão de completeza de cobertura de decisões ou de caminhos) para o algoritmo abaixo. Qual seria o valor do arrasto para o bloco de repetição associado com o while?

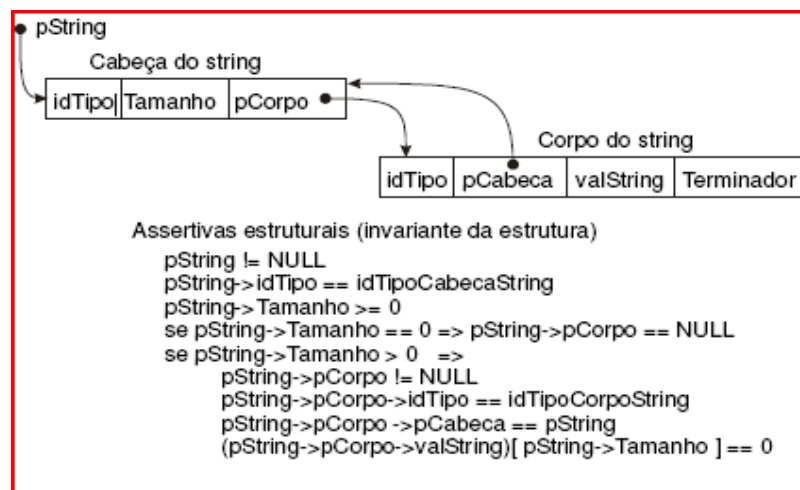
b) Descreva a argumentação do algoritmo de pesquisa binária. Não esqueça de descrever as assertivas AE, AS, AINV e as seis proposições básicas, além de apresentar uma argumentação simples da pesquisa binária.

```

// Procurar um determinado valor em um conjunto ordenado
// Inf e Sup - delimitam o domínio do conjunto no qual será realizada a pesquisa
Achou = NÃO_ACHOU ;
Inf = ObterLimInf( ) ;
Sup = ObterLimSup( ) ;
while ( Inf <= Sup )
{
    Meio      = ( Inf + Sup ) / 2 ;
    Comp      = Comparar( ValorProcurado, ObterValor( Meio ) ) ;
    if ( Comp == EH_IGUAL )
    {
        Achou = Meio ;
        break ;
    } // fim if
    if ( Comp == EH_MENOR )
    {
        Sup = Meio - 1 ; // Inf = Inf
    } else {
        Inf = Meio + 1 ; // Sup = Sup
    } // fim if
} // fim while

```

c) Esboce a função de criação de um string CriarString segundo o modelo abaixo (não se preocupe com sintaxe!). A função deve conter o código de instrumentação do controle da entrada (pre-condições, assertivas de entrada) e do resultado (pos-condições, assertivas de saída).



d) Examine o código a seguir. O código está sintaticamente correto. Quais os casos de teste que você geraria ao utilizar um critério de geração de casos de teste baseado em cobertura de arestas.

```

LER_tpCondRet LER_AbrirArquivoScript( char * NomeArqParm )
{
    strcpy( NomeArqScript , NomeArqParm ) ;
    if ( TST_ObterPtInicioExtensao( NomeArqScript ) == NULL )
    {
        strcat( NomeArqScript , DEFAULT_EXT_SCRIPT ) ;
    } /* if */
    pArqScript = fopen( NomeArqScript , "rb" ) ;
    if ( pArqScript != NULL )
    {
        printf( "\n      Arquivo de teste:      %s\n" , NomeArqParm )
        AcabouScript = 0 ;
        return LER_CondRetOK ;
    } /* if */
    NomeArqScript[ 0 ] = 0 ;
    return LER_CondRetNaoAbriu ;
} /* Fim função: LER Abrir arquivo script de teste */

```

Questao 7

Um verificador de estruturas de dados é um conjunto de uma ou mais funcoes especificamente desenvolvidas para verificar se as estruturas de dados satisfazem as suas assertivas estruturais. Responda:

- a) de 5 exemplos de funcoes verificadoras para um modulo que implementa uma estrutura generica de listas de listas.
- b) porque é dificil testar a corretude de funcoes verificadoras?

Questao 8

- a) Qual é a finalidade da instrumentacao de programas? Como redigir a instrumentacao de tal forma que ela seja facilmente retirada do modulo alvo?
- b) Como testar um validador de estruturas de dados? Nao esqueca de dar um exemplo, de preferencia usando elementos do terceiro trabalho como ilustracao.

Questao 9

O jogo Torres de Hanoi consta de um tabuleiro com tres pinos. O primeiro pino contem n discos cada qual com um diametro diferente dos demais, e ordenados de tal modo que um disco menor fica sempre sobre um disco maior. O objetivo do jogo é mover, um a um, os n discos de um pino para outro, utilizando o terceiro pino como auxiliar, sendo que jamais um disco maior podera ser colocado sobre um menor. Escreva o programa para realizar o jogo Torres de Hanoi. Argumente a corretude do programa. Mostre, passo a passo, como utilizar assertivas na construcao do modulo implementando o tipo abstrato Lista.

Questao 10

- a) Qual a diferenca entre declaracao e definicao de variaveis? Como isso ocorre com relacao a variaveis locais? E com relacao a variaveis static globais? E com variaveis publicas globais? E com ponteiros?
- b) O que é um esquema de algoritmo? De um exemplo usando um iterador capaz de, em uma arvore binaria de pesquisa, procurar por um no contendo determinado valor. A solucao deve ser generica admitindo qualquer tipo para o valor a procurar.
- c) Um documento é formado um ou mais paragrafos. Paragrafos podem ser usados tanto para textos como para titulos. Cada tipo de paragrafo define uma formatacao especifica para o texto que contem. Os atributos e operadores de formatacao de cada tipo de paragrafo sao sempre os mesmos e sao definidos em um descritor (struct) de formato de paragrafo. O texto de um paragrafo é formado por um marcador de inicio, uma ou mais linhas, cada qual contendo um string de zero ou mais caracteres, e um marcador de fim de paragrafo. Por exemplo, um paragrafo tipo X inicia com o marcador <X>, segue-se o texto do paragrafo e termina com o marcador </X>. No interior do texto de um paragrafo podem aparecer notas de rodape, identificados por <NRP>texto da nota</NRP>.
- d) Produza um modelo de dados fisico auto-verificavel para a estrutura de um documento. Nao esqueca as assertivas! Quais sao os criterios que voce utilizaria para verificar se o modelo é de fato auto-verificavel?

Questao 11

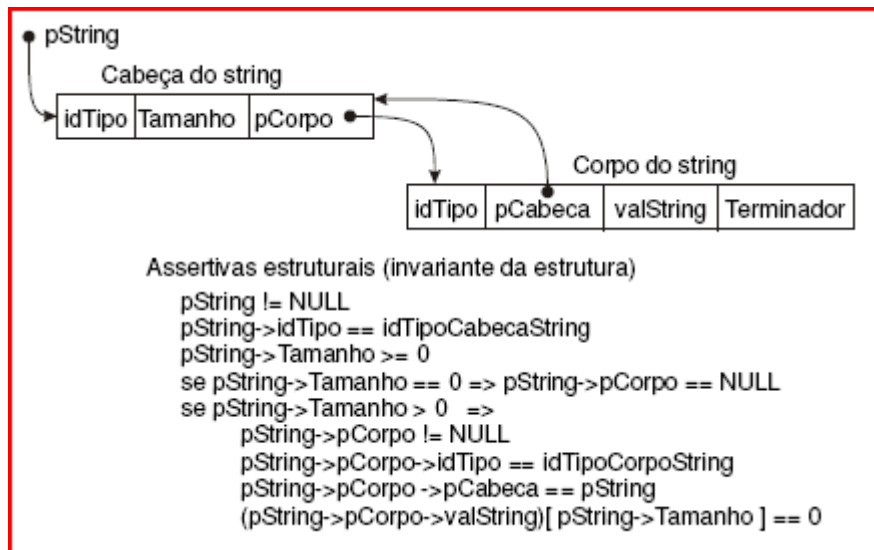
Descreva como voce geraria a partir do modelo da primeira questao os casos de teste uteis, utilizando o criterio Cobertura de Estrutura de Dados (teste de estruturas de dados). Mostre, atraves de um exemplo, como voce geraria 3 casos de teste uteis, envolvendo paragrafos.

Questao 12

Como redigir e qual a finalidade da instrumentacao de programas? Como testar um validador de estruturas de dados? Nao esqueca de dar um exemplo.

Questao 13

Crie a estrutura de decomposicao sucessiva da funcao de cadastro de turma desenvolvida no trabalho 2, apresentando os conceitos de componente concreto, componente abstrato, conjunto solucao, ortogonalidade, necessidade e suficiencia. Inclua os contadores nesta funcao segundo o padrao de completeza "cobertura de arestas". Escreva um ou mais scripts de teste capazes de verificar a cobertura dos testes.



Questao 14

a) Esboce a funcao de criacao de um string CriarString segundo o modelo acima (nao se preocupe com sintaxe!). A funcao deve conter o codigo de instrumentacao do controle da entrada (pre-condicoes, assertivas de entrada) e do resultado (pos-condicoes, assertivas de saida).

b) Por que devemos utilizar criterios de verificacao da decomposicao ao projetarmos artefatos? Descreva tres criterios que impactam diretamente a confiabilidade do artefato.

c) Esboce o deturpador do corpo do string segundo o modelo acima.

d) Que criterio voce utilizaria para escolher os casos de teste da funcao CriarString? Por que este criterio? Liste 6 casos de teste segundo este criterio.