

MC202 - Estruturas de Dados

IC - UNICAMP

Prof.: Neucimar J. Leite

Nome: NEUCIMAR

Registro acadêmico:

Turma:

12 de abril de 2013

1. (2,0) Escreva um procedimento em C para inverter uma lista ligada simples apontada por uma variável p . O seu procedimento **não pode alocar nenhum novo registro de memória**. A estrutura de dados que define um registro é dada por:

```
typedef struct RegAux {  
    int info ;  
    struct RegAux *prox ;  
} Reg, *ApReg ;
```

```
void InverteLista(ApReg *p) {  
    ApReg qq, qq1, qq2;  
    if (*p == NULL) return  
    else {  
        qq = NULL ; qq1 = *p;  
        while ( qq1 != NULL ) do {  
            qq2 = qq1->prox;  
            qq1->prox = qq;  
            qq = qq1;  
            qq1 = qq2;  
        }  
        *p = qq;  
    }  
}
```

2. (2,0) O programa a seguir recebe uma lista ligada e retorna o endereço da célula mais próxima possível do ponto médio da lista. Isto é feito sem o conhecimento do tamanho n da lista. Complete a função a seguir de modo a satisfazer este enunciado.

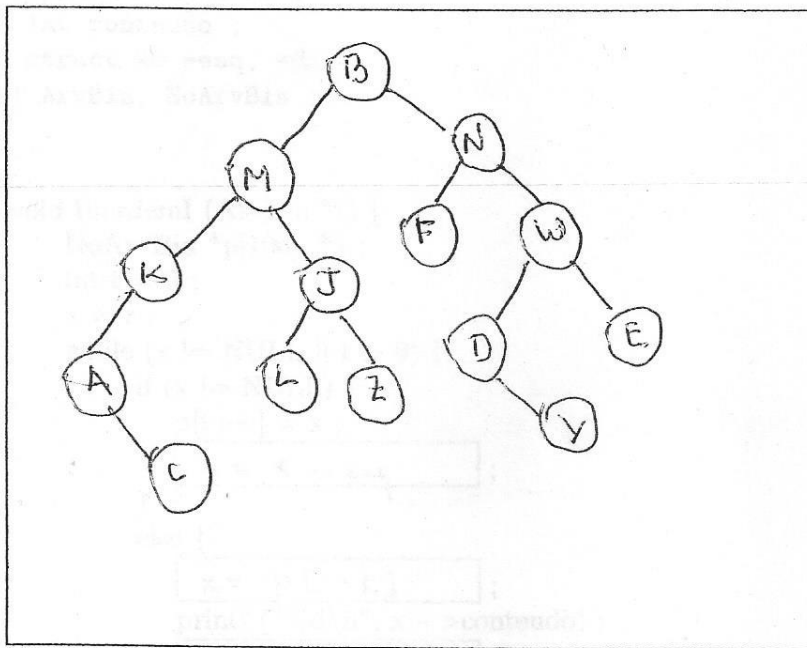
```
typedef struct No {  
    int info ;  
    struct No *prox ;  
} NoLista ;  
...
```

```
NoLista *Meio(NoLista *p) {  
    NoLista *m ;  
    m = p ;  
    while (p != NULL && p->prox != NULL) {  
        m = m->prox ;  
        p = p->prox->prox ;  
    }  
    return m ;  
}
```

3. (1,5) (a) Reconstrua a árvore binária a partir dos seus percursos em pós-ordem e inordem:

Pós-ordem: C A K L Z J M F Y D E W N B

Inordem: A C K M L J Z B F N D Y W E



(b) (0,5) Indique o percurso em pré-ordem para a árvore obtida acima:

B M K A C J L Z N F W D Y E

5. (2,0) Dê um nome adequado a cada uma das funções XXX, YYY, ZZZ e WWW abaixo:

(a) (1,0) Função YYY:

```
int YYY (ArvBin *p) {  
    int n,m ;  
    if (p == NULL) return 0 ;  
    else {  
        n = YYY(p->esq) ;  
        m = YYY(p->dir) ;  
        if (n < m) return (1 + m) ;  
        else return (1 + n) ;  
    }  
}
```

altura da árvore

(b) (1,0) Função WWW:

```
boolean WWW (ArvBin *p1, ArvBin *p2) {  
    if ((p1 == NULL) && (p2 == NULL)) return TRUE ;  
    else  
        if (p1 == NULL) return FALSE ;  
        else  
            if (p2 == NULL) return FALSE ;  
            else  
                if (p1->info == p2->info)  
                    return ((WWW(p1->esq, p2->esq) && WWW(p1->dir, p2->dir)) ;  
                else  
                    return FALSE ;  
}
```

compara duas árvores

4. (2,0) A função abaixo percorre iterativamente uma árvore binária apontada por p , em *inordem*, imprimindo o conteúdo de cada um de seus nós. Ela supõe um máximo de 100 nós na árvore. Complete os espaços a seguir de modo a satisfazer este enunciado.

```
typedef struct No {  
    int conteudo ;  
    struct No *esq, *dir ;  
} ArvBin, NoArvBin ;  
...
```

```
void InordemI (ArvBin *r) {  
    NoArvBin *p[100], *x ;  
    int t = 0 ;  
    x = r ;  
    while (x != NULL || t > 0) {  
        if (x != NULL) {  
            p[t++] = x ;  
             $x = x \rightarrow \text{esq}$  ;  
        }  
        else {  
             $x = p[--t]$  ;  
            printf ("%d\n", x->conteudo) ;  
             $x = x \rightarrow \text{dir}$  ;  
        }  
    }  
}
```