

INF 1620 – P2 - 30/05/03	Questão 1
Nome:	
Matrícula:	Turma

Considere a implementação de uma lista encadeada para armazenar as notas dos alunos de uma turma. O tipo `Lista` representa a estrutura do nó da lista conforme declaração abaixo:

```
typedef struct lista Lista;
```

Pede-se:

- Defina `struct lista` de tal forma que a informação associada a cada nó da lista seja composta por:
  - Nome do aluno: cadeia com até 80 caracteres
  - Nota do aluno: número real
- Escreva uma função para inserir um novo elemento na lista. A ordem em que o elemento será inserido na lista é irrelevante. Esta função deve ter como valor de retorno o ponteiro para a lista alterada e deve obedecer o seguinte protótipo:  
`Lista* insere (Lista* l, char* nome, float nota);`

*A estrutura do nó da lista pode ser dada por:*

```
struct lista {  
    char nome[81];  
    float nota;  
    struct lista* prox;  
};
```

*Uma função para inserir no início da lista pode ser dada por:*

```
Lista* insere (Lista* l, char* nome, float nota){  
    Lista* p = (Lista*) malloc(sizeof(Lista));  
    strcpy(p->nome, nome);  
    p->nota = nota;  
    p->prox = l;  
    return p;  
}
```

---

Funções da biblioteca padrão de C que podem ser úteis:

```
int strlen (char* s);  
char* strcpy (char* s1, char* s2);  
char* strcat (char* s1, char* s2);
```

*Não separe as folhas deste caderno. Todas as folhas devem ter seu nome. Responda cada questão na folha correspondente. Use o verso se necessário.*

INF 1620 – P2 - 30/05/03	Questão 2
Nome:	
Matrícula:	Turma

Considerando uma lista encadeada de valores inteiros definida pelo tipo abaixo:

```
struct lista {
    int info;
    struct lista* prox;
};
typedef struct lista Lista;
```

Escreva uma função que retire o último elemento de uma dada lista. Esta função deve ter como valor de retorno o ponteiro para a lista alterada e deve obedecer o seguinte protótipo:

```
Lista* retira_ultimo (Lista* l);
```

*Obs:* o caso da lista vazia deve ser considerado.

*Uma função para retirar o último elemento da lista pode ser dada por:*

```
Lista* retira_ultimo (Lista* l) {
    Lista* p = l;
    Lista* ant = NULL;

    if (p == NULL)
        return NULL;

    while (p->prox != NULL) {
        ant = p;
        p = p->prox;
    }

    if (ant == NULL)
        l = NULL;
    else
        ant->prox = NULL;

    free(p);
    return l;
}
```

*Não separe as folhas deste caderno. Todas as folhas devem ter seu nome. Responda cada questão na folha correspondente. Use o verso se necessário.*

INF 1620 – P2 - 30/05/03	Questão 3
Nome:	
Matrícula:	Turma

Considere a existência de um tipo abstrato `Pilha` de números reais, cuja interface é definida pelo arquivo `pilha.h` com o seguinte conteúdo:

```
typedef struct pilha Pilha;

Pilha* cria (void);          /* cria uma pilha vazia */
void push (Pilha* p, float v); /* insere um elemento no topo */
float pop (Pilha* p);       /* retira o elemento do topo */
int vazia (Pilha* p);       /* retorna se está vazia */
void libera (Pilha* p);     /* libera a pilha */
```

Sem conhecer a representação interna deste tipo abstrato e usando apenas as funções declaradas no arquivo `pilha.h`, implemente uma função que imprima os valores dos elementos armazenados numa pilha dada. Os elementos devem ser impressos da base para o topo, isto é, o elemento do topo deve ser o último a ser impresso. Ao final da função, a pilha deve ter o mesmo conteúdo original. Esta função deve obedecer o seguinte protótipo:

```
void imprime (Pilha* p);
```

*Dica:* use uma pilha auxiliar para transferir os valores da pilha original ou faça uma implementação recursiva.

*Usando uma pilha auxiliar:*

```
void imprime (Pilha* p) {
    /* transfere elementos para pilha auxiliar */
    Pilha* aux = cria();
    while (!vazia(p)) {
        push(aux, pop(p));
    }
    /* transfere de volta imprimindo valores */
    while (!vazia(aux)) {
        float v = pop(aux);
        printf("%f\n", v);
        push(p, v);
    }
    libera(aux);
}
```

*Usando recursão:*

```
void imprime (Pilha* p) {
    if (!vazia(p)) {
        float v = pop(p);
        imprime(p);
        printf("%f\n", v);
        push(p, v);
    }
}
```

*Não separe as folhas deste caderno. Todas as folhas devem ter seu nome. Responda cada questão na folha correspondente. Use o verso se necessário.*

INF 1620 – P2 - 30/05/03	Questão 4
Nome:	
Matrícula:	Turma

Numa estrutura de árvore, o nível (ou altura) de um nó é definido como sendo igual à distância da raiz até o nó em questão. Assim, o nó raiz está no nível 0, os filhos da raiz no nível 1, os netos no nível 2, e assim por diante.

Considerando uma estrutura de árvore binária de busca definida pelo tipo abaixo:

```
struct arv {
    int info;
    struct arv *esq, *dir;
};
typedef struct arv Arv;
```

Escreva uma função que retorne o nível do nó cuja informação seja igual a um valor  $x$  dado. Esta função deve tirar proveito da ordenação dos nós da árvore de busca, onde as informações associadas aos nós da sub-árvore à esquerda são menores que a informação associada à raiz e as da sub-árvore à direita são maiores. Se a árvore for vazia, a função deve ter como valor de retorno o valor  $-1$ , caso contrário, pode-se assumir que um único nó com valor  $x$  sempre estará presente na árvore..

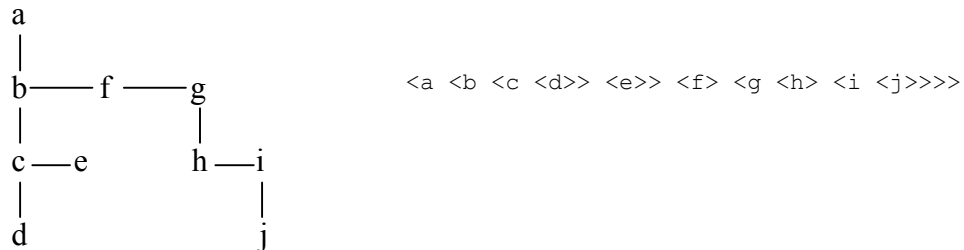
```
int nivel (Arv* a, int x);

int nivel (Arv* a, int x) {
    int n;
    if (a==NULL)
        return -1;
    else if (x==a->info)
        return 0;
    else {
        int n;
        if (x<a->info)
            n = nivel(a->esq,x);
        else
            n = nivel(a->dir,x);
        if (n >= 0)
            return n+1;
        else
            return -1;
    }
}
```

*Não separe as folhas deste caderno. Todas as folhas devem ter seu nome. Responda cada questão na folha correspondente. Use o verso se necessário.*

INF 1620 – P2 - 30/05/03	Questão 5
Nome:	
Matrícula:	Turma

O conteúdo de uma árvore genérica pode ser representada de forma textual pela seguinte notação: <raiz sa<sub>1</sub> sa<sub>2</sub> ... sa<sub>n</sub>>. A figura abaixo ilustra esta representação textual de uma árvore genérica tomada como exemplo.



Considerando a estrutura de árvore genérica que armazena caracteres definida pelo tipo abaixo:

```
struct arvgen {
    char info;
    struct arvgen *prim; /* ponteiro p/ primeiro filho */
    struct arvgen *prox; /* ponteiro p/ irmão */
};
typedef struct arvgen ArvGen;
```

Escreva uma função que imprima o conteúdo da árvore usando a representação textual mostrada acima. O protótipo da função é dado por:

```
void imprime (ArvGen* a);

void imprime (ArvGen* a) {
    ArvGen* p;
    printf("< %c ", a->info);
    for (p=a->prim; p!=NULL; p=p->prox)
        imprime(p);
    printf(">");
}
```

*Não separe as folhas deste caderno. Todas as folhas devem ter seu nome. Responda cada questão na folha correspondente. Use o verso se necessário.*

## RASCUNHO

*Respostas nesta folha não serão consideradas.*

*Não separe as folhas deste caderno.*