

```
// IED-001 (Prof. Dr. Silvio do Lago Pereira)

// -----
// Exemplo 1
// -----

typedef int Item;
typedef struct no {
    Item item;
    struct no *prox;
} *Lista;

// -----
// Exemplo 2
// -----

Lista no(Item x, Lista p) {
    Lista n = malloc(sizeof(struct no));
    n->item = x;
    n->prox = p;
    return n;
}

// -----
// Exemplo 4
// -----

void exhibe(Lista L) {
    while( L != NULL ) {
        printf("%d\n",L->item);
        L = L->prox;
    }
}

// -----
// Exercício 1
// -----

#include <stdio.h>
#include <stdlib.h>

typedef int Item;
typedef struct no {
    Item item;
    struct no *prox;
} *Lista;

...

int main(void) {
    Lista I = no(3,no(1,no(5,NULL)));
    exhibe(I);
    return 0;
}

// -----
// Exemplo 5
// -----

int tamanho(Lista L) {
    int t = 0;
    while( L ) {
        t++;
        L = L->prox;
    }
    return t;
}
```

```
}

// -----
// Exercício 3
// -----

#include <stdio.h>
#include <stdlib.h>

...

int main(void) {
    Lista I = no(3,no(1,no(5,NULL)));
    exhibe(I);
    printf("Tamanho = %d\n",tamanho(I));
    return 0;
}

// -----
// Exemplo 6
// -----

Lista aleatoria(int n, int m) {
    Lista L = NULL;
    while( n>0 ) {
        L = no(rand()%m, L);
        n--;
    }
    return L;
}

// -----
// Exercício 5
// -----

#include <stdio.h>
#include <stdlib.h>
#include <time.h>

...

int main(void) {
    srand(time(NULL));
    Lista A = aleatoria(10,100);
    exhibe(A);
    return 0;
}

// -----
// Exemplo 7
// -----

void anexa(Lista *A, Lista B) {
    if( !B ) return;
    while( *A )
        A = &(*A)->prox;
    *A = B;
}

// -----
// Exercício 7
// -----

#include <stdio.h>
```

```
#include <stdlib.h>
```

```
...
```

```
int main(void) {  
    Lista H = no(4,no(2,NULL));  
    Lista I = no(3,no(1,no(5,NULL)));  
    printf("H = "); exibe(H);  
    printf("I = "); exibe(I);  
    printf("Pressione enter");  
    getchar();  
    anexa(&H,I);  
    printf("H = "); exibe(H);  
    printf("I = "); exibe(I);  
    return 0;  
}
```

```
// -----  
// Exemplo 8  
// -----
```

```
void destroi(Lista *L) {  
    while( *L ) {  
        Lista n = *L;  
        *L = n->prox;  
        free(n);  
    }  
}
```