

Obs. Nas questões abaixo, suporemos que temos

```
typedef struct node *link;
struct node { int item; link next; };
```

1. [3 pontos] Escreva uma função de protótipo

```
link elem_max(link h);
```

que recebe como entrada um ponteiro para o primeiro elemento de uma lista, e devolve um ponteiro para um elemento desta lista que tem campo `item` máximo. Se a lista dada for vazia (isto é, se `h` for `NULL`), a função deve devolver `NULL`. Suponha que não temos nem cabeça nem cauda em nossas listas.

2. [3 pontos]

(i) Escreva uma função de protótipo

```
link copia(link h);
```

que recebe um ponteiro para o começo de uma lista e devolve um ponteiro para uma nova lista, que é uma cópia da lista dada. A nova lista deve conter uma cópia de cada elemento da lista original; estas cópias devem ocorrer 'na mesma ordem' que na lista original. Suponha que não temos nem cabeça nem cauda em nossas listas.

(ii) Escreva uma função de protótipo

```
link copia_reversa(link h);
```

que recebe um ponteiro para o começo de uma lista e devolve um ponteiro para uma nova lista, que é uma cópia do *reverso* da lista dada. A nova lista deve conter uma cópia de cada elemento da lista original; estas cópias devem ocorrer 'na ordem inversa' da lista original. Suponha que não temos nem cabeça nem cauda em nossas listas.

3. [4 pontos] Nesta questão, supomos que todas as listas têm uma mesma cauda, a saber, o elemento `z` com `z->next = z`. Em particular, a lista 'vazia' (com '0 elementos') contém o elemento `z`. Considere a seguinte função:

```
link misterio(link *aa, link *bb, link c)
{ link a, b;
  *aa = *bb = z;
  if (c->next != z) {
    a = c; b = c->next->next->next;
    while (b != z)
      { c = c->next; b = b->next->next; }
    b = c->next; c->next = z;
    *aa = a; *bb = b;
  }
  return z;
}
```

Uma chamada da função acima em uma lista de começo `c` cria três listas, dadas pelo valor devolvido pela função, por `*aa` e por `*bb`.

- (i) Determine o número de elementos que estas três listas terão quando a lista original (dada por `c`) tem, respectivamente, 0, 1, 2, 3, 4, 5, e 6 elementos. Em todas as contagens, *não* conte a cauda!
- (ii) Idem para o caso geral, em que a lista original tem $n \geq 0$ elementos. Justifique sua resposta sucintamente.