1. Escreva uma função que calcula o número de folhas em uma árvore dada.

```
int folhas (p_no raiz) {
  if (raiz == NULL)
    return 0;
  if (raiz->esq == NULL && raiz->dir == NULL)
    return 1;
  return folhas(raiz->esq) + folhas(raiz->dir);
}
```

2. Escreva uma função recursiva que apaga todas as folhas de uma árvore que tenham a chave igual a um valor dado.

```
p_no apaga_folhas(p_no raiz, int chave) {
  if (raiz == NULL)
    return raiz;
  if (raiz->esq == NULL && raiz->dir == NULL && raiz->chave == chave) {
    free(raiz);
    return NULL;
  }
  raiz->esq = apaga_folhas(raiz->esq, chave);
  raiz->dir = apaga_folhas(raiz->esq, chave);
  return raiz;
}
```

3. Escreva uma função que compara se duas árvores binárias de busca são iguais.

```
int compara(p_no raiz1, p_no raiz2) {
  if (raiz1 == NULL && raiz2 == NULL)
    return 1;
  if (raiz1 != NULL && raiz2 == NULL)
    return 0;
  if (raiz1 == NULL && raiz2 != NULL)
    return 0;
  /*neste ponto nenhuma eh NULL*/
  return raiz1->chave == raiz2->chave &&
    compara(raiz1->esq, raiz2->esq) &&
    compara(raiz1->dir, raiz2->dir);
}
```

4. Faça uma implementação da função `sucessor` que não usa o ponteiro `pai`

- Dica: você precisará da raiz da árvore pois não pode subir

```
/*supondo no != NULL*/
p_no sucessor(p_no no, p_no raiz) {
  if (no->dir != NULL)
    return minimo(no->dir);
  return ancestral_a_direita(no, raiz);
}

p_no ancestral_a_direita(p_no no, p_no raiz) {
  p_no ancestral == NULL;
  while(atual != no) {
    if (no->chave < atual->chave) {
      ancestral = atual;
      atual = atual->esq;
    }
    else
      atual = atual->dir;
  }
  return ancestral;
}
```

5. Escreva uma função em C que, dado uma lista de chaves inteiras em um vetor, encontra inteiros positivos a e M com M mínimo tal que a função de hashing $a \cdot x \bmod M$ é injetora para a lista de chaves dada, isto é, qualquer duas chaves diferentes x e y (da lista dada) temos que $a \cdot x \bmod M \neq a \cdot y \bmod M$.

```
int viavel(int *v, int n, int M, int a) {
  for (i = 0; i < n; i++)
    for (j = i+1; j < n; j++)
      if (a*v[i]%M == a*v[j]%M)
        return 0;
  return 1;
}

void encontra_hashing(int *v, int n, int *M, int *a) {
  for(*M = 2; ; *M++)
    for(*a = 1; *a < *M; *a++)
      if (viavel(v, n, *M, *a))
        return;
}
```