```
#include <iostream>
using namespace std;

struct Node {
  int val;
  Node *next;
};

struct List {
  Node *head;
};

bool checkCircular(Node *n, Node *frt) {
  if(!n) {
    return false;
  } else if(n == frt) {
    return true;
  } else {
    return checkCircular(n->next, frt);
  }
}

bool isCircular(const List &l) {
  if(!l.head) return false;
  else return checkCircular(l.head->next, l.head);
}

int findMinCur(Node *cur, int minsofar) {
  if(!cur) return minsofar;

  if(minsofar > cur->val) minsofar = cur->val;
  return findMinCur(cur->next, minsofar);
}

int findMin(const List &l) {
  if(!l.head) exit(1);
  return findMinCur(l.head->next, l.head->val);
}
```

```
int recursivePower(int base, int exponent) {
  if(exponent == 0) return 1;
  else if(exponent == 1) return base;
  else if(exponent % 2 == 0) {
    int x = recursivePower(base, exponent / 2);
    return x * x;
  } else return recursivePower(base, exponent / 2) * recursivePower(base, exponent / 2 +
}
```