

1. (25 Points)

Using only the List ADT operations defined in the project description for pa1 (pages 2 and 3), write a client function with the heading

```
int searchList(List L, int x)
```

Your function will perform a linear search in L for the target x . Your function will return the lowest index (i.e. nearest the front) at which the target is found, if it is found, and will return -1 if the target is not found. Upon return, the input list will have its cursor under the found element, if found, and will have its cursor in the undefined state if the target is not found.

One of several possible solutions:

```
int searchList(List L, int x){
    for(moveFront(L); index(L)>=0; moveNext(L)){
        if( x==get(L) ) break;
    }
    return( index(L)>=0 ? index(L) : -1 );
}
```

2. (25 Points)

Using only the List ADT operations defined in the project description for pa1, (pages 2 and 3), write a client function with the heading

```
int isPalindrome(List L)
```

Your function will return `true` (1) if the integer sequence represented by L is a palindrome (i.e. is identical to its own reversal.) Your function will return `false` (0) if L is not a palindrome.

One of several possible solutions:

```
int isPalindrome(List L){
    int eq;
    List R = newList();
    for(moveFront(L); index(L)>=0; moveNext(L)){
        prepend(R, get(L));
    }
    eq = equals(L, R);
    freeList(&R);
    return eq;
}
```