

**Problem 1** 40 points

(a) i.next

(b) j->next->value

(c) \*k

(d) (\*k)->next

- a) struct n \* / node \* / pointer of node  
b) integer  
c) struct n \* / node \* / pointer of node  
d) struct n \* / node \* / pointer of node

**Problem 2** 30 points

```
// This function will compare the value field of each  
// element of L1 and L2 and returns 1 if they match,  
// and 0 otherwise.
```

```
int compare (node * L1, node * L2) {  
    while (L1 != NULL && L2 != NULL) {  
        if (L1->value != L2->value)  
            return 0;  
        L1 = L1->next;  
        L2 = L2->next;  
    }  
    if (L1 != NULL || L2 != NULL)  
        return 0;  
    return 1;  
}
```

**Problem 3** 30 points

```
int IsEmpty (node *list); /* returns 0 if list is not empty, */  
/* 1 if it is empty. */  
int First (node *list); /* returns value of first node of list */  
node *Rest (node *list); /* returns pointer to second element of */  
/* list. If list has only one item, */  
/* return NULL. */
```

Write a recursive function which adds up the value fields of the nodes in the list parameter using the above three functions.

```
int AddEm (node *list) {  
    if (IsEmpty(list))  
        return 0;  
    return First(list) + AddEm(Rest(list));  
}
```