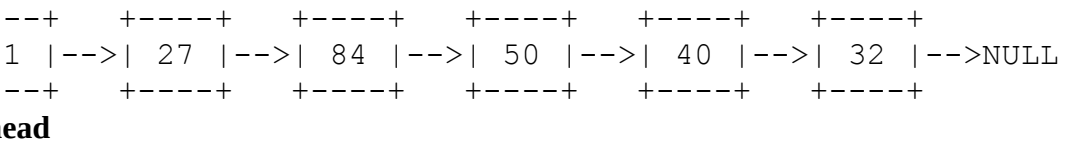
2) (5 pts) DSN (Linked Lists)

Consider the following function, which takes the head of a linked list as its only input parameter:

```
node *funky(node *head) {
  if (head == NULL)
    return head;
  if (head->next != NULL && (head->next->data % 2) == 0) {
    head->next = yknuf(head->next->next, head->next);
    head = funky(head->next->next);
  }
  else if (head->next != NULL)
    head->next = funky(head->next);
  return head;
}

node *yknuf(node *n1, node *n2) {
  n2->next = n1->next->next;
  n1->next = n2;
  return n1;
}
```

Suppose someone passes the head of the following linked list to the *funky()* function:



The function call is: `funky(head);`

This program is going to crash spectacularly, but before it does, it will change the structure of the linked list a bit. Trace through the function call(s) and draw a new diagram that shows how the links in this linked list will be arranged at the moment when the program crashes. (In particular, show where the next pointer for each node except the one storing 32 will point.)