doesn't work correctly in all cases. Explain what's wrong with it and show how to fix it. Assume that the node structure is the one defined in Section 17.5.

```
struct node *insert_into_ordered_list(struct node *list,
                                      struct node *new_node)
{
  struct node *cur = list, *prev = NULL;
  while (cur->value <= new_node->value) {
    prev = cur;
    cur = cur->next;
  }
  prev->next = new_node;
  new_node->next = cur;
  return list;
}
```

Section 17.6 14. Modify the delete_from_list function (Section 17.5) so that its first parameter has type struct node ** (a pointer to a pointer to the first node in a list) and its return type is void. delete_from_list must modify its first argument to point to the list after the desired node has been deleted.

Section 17.7 ⓦ 15. Show the output of the following program and explain what it does.

```
#include <stdio.h>

int f1(int (*f)(int));
int f2(int i);

int main(void)
{
  printf("Answer: %d\n", f1(f2));
  return 0;
}

int f1(int (*f)(int))
{
  int n = 0;

  while ((*f)(n)) n++;
  return n;
}

int f2(int i)
{
  return i * i + i - 12;
}
```

16. Write the following function. The call sum(g, i, j) should return g(i) + ... + g(j).

```
int sum(int (*f)(int), int start, int end);
```

ⓦ 17. Let a be an array of 100 integers. Write a call of qsort that sorts only the *last* 50 elements in a. (You don't need to write the comparison function).

18. Modify the compare_parts function so that parts are sorted with their numbers in *descending* order.

19. Write a function that, when given a string as its argument, searches the following array of structures for a matching command name, then calls the function associated with that name.