

1. Log into `linux.grace.umd.edu` and copy the compressed tarfile `discussion16.tgz` from the directory `212public/discussions` to your extra disk space using the symbolic link `212public`.
2. Extract the files using `gtar -zxvf discussion16.tgz`, which will create a subdirectory `discussion16`, so `cd` there. You'll find a program skeleton there named `lists.c`.
3. The file has four functions that you are to complete. The effects of the functions are described below.

After writing each function below, compile the program and test it to ensure it doesn't have any errors. (Note that since the program is contained in a single file, we didn't bother to create a makefile.)

Also add other calls to test the functions in different cases, and test them with different list contents! The program has a function `create_list()` that just creates a small list with the three values 10, 20, and 30, used to test the functions you are to write. You may want to write other versions of the `create_list()` function, which create lists of different lengths, for testing purposes.
4. `delete_second()` should remove the second element from a singly-linked list. If the list doesn't have at least two elements it should have no effect.
5. `sum_odd()` should return the sum of all the values stored in the odd-numbered nodes of the list (the first node, the third node, etc.). The function should work correctly without errors regardless of how many elements are in the list.
6. `sorted()` should return 0 or 1 after determining whether all the values in a singly-linked list appear in ascending sorted numerical order (0 if the list is not sorted, 1 if it is). The function should work correctly without errors regardless of how many elements are in the list.
7. `reverse()` should reverse the order of the elements (`Nodes`) in its parameter list. The function should work correctly without errors regardless of how many elements are in the list.
8. As mentioned, after writing each function, add other calls to test it in different cases! Test them with different list contents, such as an empty list, a list with one element, a list with two elements, a list with more than three elements, lists with different values, etc.
9. Don't forget to submit what you've done by the end of your discussion section, whether it works right or not, to get credit for the exercise.