

CS29206: Systems Programming Lab
Spring 2025
Assignment 4

Please follow instructions carefully.

1. Create a directory named **A4<your roll no>** under your home directory (directory you are in when you log in). For example, if your roll no. is 23CS10002, the directory should be named **A423CS10002**. All instructions after this in this assignment will use this directory name, **A423CS10002**, as example, you should interpret it as your actual directory name you will use based on your own roll no.
2. Copy the assignment tar file for Assignment 4 (**Assignment4.tar**) in your directory. The tar file contains a C file named **list.c**, plus this assignment pdf. Unzip the file. The C file contains a complete C program to add and delete positive integers to a list that is essentially a doubly-linked list. The function names are self-explanatory.
3. Compile the file **list.c** file to create an executable file **a.out**.
4. Run **a.out** with **valgrind** with full memory leak check. You should store the output of **valgrind** to a file named **leak.txt**. You can specify this to valgrind using the flag **--log-file="leak.txt"** in the **valgrind** command line.
5. The program will ask for the last 5 digits of your roll no. Based on your roll no., it will add a certain number of elements to the list and then delete them. It is easy to see from the code what is being added and deleted and the order of addition/deletion. The list created after adding all elements is printed on the screen (in descending order).
6. Open the **leak.txt** file. Cut and paste the list elements printed on the screen to the end of the **valgrind** output (give exactly 3-line gap after the **valgrind** output ends). Then answer each of the following questions in the same file (Write the question no. like (a), (b), ... Give 2-line gaps between answers to each question).
 - (a) How many bytes of memory leak is reported and of what types? No explanation needed.
 - (b) What is the meaning of a "block" in the **valgrind** output you see? One sentence only.
 - (c) Based on your understanding of what the code is doing, can you suggest a different order of deletion of the elements in the list (for the same integers added in the same order as done in the program) that will reduce the leak to a minimum? You can try out other deletion orders by changing ONLY the **main()** function in **list.c**, but finally write only the order that will give the minimum leak. Write the order as the list of numbers in the list in the order of deletion, and then in the next line, write 1 sentence ONLY what you think best characterizes the order (i.e., gives a general description of the deletion order that will give the minimum leak). You need to understand what the code is doing a bit for this.
 - (d) Explain in 2-4 sentences max. why the leak is happening, specifically referring to line numbers of the code in the **list.c** file given.

7. Modify the C file to fix the code so that there is no memory leak irrespective of the order of addition and deletion and store the new program in a file called *mylist.c*. Do not change any function names. Add a comment “Beginning of change done” before every change (add/modify) done and add a comment “End of change done” after every change. Any change done that is not encased by comments around it will be ignored. Do NOT change the nature of the list or the logic of adding/deleting. The change you have to do is actually very small.
8. Tar/zip the two files *leak.txt* and *mylist.c* and upload in moodle in the submission link for Assignment 4.