# Advanced Programming
# COEN 11

## Lab 6

# Lab6

## Spreadsheet

- Change lab 5 to use an array of  lists
  - There are 3 lists according to the age group
    - <= 18
    - > 18 and < 65
    - >= 65
- Due in week 7

# Lab6

## Adding to the functionality of Lab 3

- The spreadsheet is created interactively with the following commands
  - 1
    - Insert at the tail of the right list
  - 2, 3
    - Traverse the lists starting from head
  - 4 <first name> <last t name>
    - Remove the item with the first and last names given
  - 0
    - Save the info to the file, delete (free) all the nodes, and quit

# Lab6

## Main Requirements

- Main: loop forever accepting commands
  - Use a switch  statement
- Do not allow names to repeat
- 8 functions
  - main, insert, show, show_age, delete, check_duplicate, save_all, read_all
- List mechanism
  - Your lists should stay in an old-to-new order
    - Always insert a new entry at the tail

# Lab6

## More requirements

- Global variable, just one!
  - Array with 3 structs
    - The struct should have a head and  a tail

# Lab 6

## Requirements

- Define a struct list
  - head and tail
- Array of struct list, size = 3
  - heads and tails need to be initialized to NULL
- Save the info and free all the nodes before quitting
  - New function for option zero: save info and delete nodes

# Lab 6

## Extra Credit (10 points on the 1st midterm)

- Add an option to change the age, given a name
  - 6 <first name> <last name> <old_age> <new_age>
    - Traverse the old_age list searching for name
    - If found, the info moves to the end of the new_age list
      - » Delete the node and call insert

# Lab 6

## Saving the info to a file

- Add saving and retrieving

# Lab 6

## Initially

- The lists may be either
  - Empty
  - Formed with information read from a file

## At the end

- The updated lists are saved into a file

# Lab 6

The info should be saved in a text file according to the following format:

| | | |
|---|---|---|
| Joe | Smith | 5 |
| Mary | Miller | 7 |
| Zoe | Lopez | 20 |
| Ann | Chen | 67 |

It should be possible to read the file with commands such as cat and more

# Lab 6

The name of the file is an argument for the program

- If the file does not exist
  - fopen returns NULL for reading
  - the lists start empty and are saved at the end into a file with the given name
- If the file does exist
  - the lists are initially formed with the information obtained from the file and is saved into the same file at the end

# Lab 6

The name of the file is an argument for the program

- Example:

  # ./lab6  file_name

  or

  # ./a.out  file_name

# Lab 6

The name of the file is the first argument for the program

- In the code:

```
main (int argc, char *argv[ ])
{
   . . .
   if (argc == 1)
   {
      printf (“The name of the file is missing!\n”);
      return 1;
   }

   read_file (argv[1]);
   …
```

# Lab 6

The name of the file is an argument for the program

- In the code:
  - argc gives the number of arguments
  - argv is an array of strings, each of which is one of the arguments for the program
  - argv[0] is the name of the executable
  - argv[1] – argv[argc – 1] are the arguments

# Lab 6

The spreadsheet is created/modified interactively, except that command quit (zero) will save the info into a file

- quit
  - save the list in the file specified
  - delete each node as they are saved
  - quit

# Lab 6

## More Requirements

- Two new functions, called from main
  - Read from file
    - Receive file name as an argument
    - Call insert to insert the data read from file
  - Save to file
    - Receive file name as an argument

# Lab 6

## More Requirements

- Use same insert function for inserting information from the file and from the keyboard.
- Your insert function should have the following type:

  ```
  void insert (char *, char *, int);
  ```

- Read the names and age to local variables (char array, char array, int) before calling the insert function.

# Lab 6

## More Requirements

- Names cannot repeat!
  - Need to deal with that before calling function insert

# Lab 6

## To receive full credit

- Pre-lab (10%)
  - Test plan
- Demo (30%)
  - Show the TA
    - Start with an empty list
    - Add two people to to each list
    - Show each command
    - Quit
    - Start again
    - Show the list
- Submit to Camino (60%)