

Module 8 – Introduction to Linked Lists

Tutorial Questions

Objectives

To understand some of the concepts relating to dynamic memory allocation and dynamic memory structures in C. To understand the important concepts involved in designing dynamic data structures (structures that can grow), using a linked list as a first example of this. These skills will be important in your implementation of assignment 2.

1. What are the dynamic memory allocation functions in C? How do we use them and where is this memory allocated from?

Consider the following data structure:

```
#define NAMELEN 40
struct person
{
    char fname[NAMELEN+1];
    char lname[NAMELEN+1];
    unsigned int age;
};
typedef struct person Person;
```

2. List the function prototypes for initialisation, adding to a list, deleting an item and freeing the list when we are done with it. What about printing all elements in the list?
3. A linked list is a dynamic data structure that has the structure on the following page. Write down some C code that declares this data structure. Make the data node a pointer to a Person. Hint: There will be two structs.
4. Implement the List Initialisation function you listed the prototype for earlier. What needs to get initialised here?
5. Implement the function for adding a (Person *) to the list. You will need to insert them in sorted order by their last name. Are there any special cases you need to be aware of?
6. Implement a main function which prompts the user to enter a series of “Persons”. You will need to declare a list and initialise it. You will need to allocate the memory for each person before calling the appropriate function to insert this user in the list. If the user presses enter at a new line we know they have finished entering data. Finally, call the print and free functions that were listed under the prototypes. We will look at the implementation of these next week.

The Linked List Data Structure

