# CIS\*2500 W20 Assignment 4 Questions & Answers Part 7

**“Is nth 0 or 1 indexed? In other words does it start from 0 or from 1?”**

It is 0 indexed. The example input/output provide a demonstration of its use.

Mark Wineberg

**“On Q1b with the function map\_2\_reduce I noticed there is no value\_type init parameter like the other reduce functions. Is this intentional? If so are we supposed to assume the initial value is 0?”**

No, it was a change I made to the reduce function that got applied to map\_reduce and the lecture notes, but map\_2\_reduce was missed.

The assignment has now been updated to make all "reduces" consistent.

Mark Wineberg

**“ The print\_list\_all and print\_list\_sort functions has to be different for value\_types of int, char [80] and Fraction. If we place these functions in sort\_list.c, how can we handle this? Could we use #ifdef here as well?”**

In q1a, it was not specified that print\_list\_all and print\_list\_sort be in sort\_list.c. There was two variants, which function differently. So its signature is not mandatory in general, and can be changed for q3.

That being said, if you have put the function in sort\_list.c, then you have to have only one version of the functions as there can only be one sort\_list.c. This can be done in one of two ways:

1) use #ifdef as done for assignments and typedefs using -DFRACT.

2) create a general version of print\_list\_all and print\_list\_sort using function pointers to pass in how the node should be printed. Then have two different ways of printing the fraction, a simple way and a mixed way and pass in the appropriate function as needed.

The second option is the better one in general, but is requires an understanding of function pointers, which is covered in q1b, not q1a, which is why I didn't demand it.

Mark Wineberg

**“As value\_type is defined, for a4q1a\_char as**

**typedef char value\_type[80];**

**Then it cannot compile with the reduce or map\_reduce functions in q1b as you cannot return value\_type of this type.**

**value\_type reduce(...);**

**cannot return arrays, only a pointer to an array which would have a type set by**

**typedef char \* value\_type;**

**which is different.**

**How should this be handled?”**

You do not have to generalize the map/reduce functions to cover the char [80] case.

The reason follows:

In general, reduce should have its own "reduce\_type" that it returns. The same type would be used for the init argument passed in.

This applies in general. Map should actually have an entire Map\_List type, one for each type of map function that could be applied. This begins to be unworkable.

The solution to this is for map and reduce to use *void pointers* in its function pointer definition / init arg / return value. This is completely general and has the added benefit of not needing recompilation of the list library every time you use a different application. The void pointer notes will be released tomorrow (Monday), so you don't know about them yet, but will.

I purposely didn't include these ideas in the assignment design to prevent the need to understand that I didn't cover live in class, and which would have affecting all questions, including q1a .

Consequently, the assignment was specifically designed so that char [80] only showed up in q1a to avoid the problem you are having with it when applied to reduce and map\_reduce.

Mark Wineberg

**“Sir in the input/output example for Fraction, you are giving numbers bigger than the long data type can see”**

You are right ... sort of.

The answer is, it is compiler dependent  (and the "same" compiler can actually work differently on different machines).

Some compilers treat long as 8 bytes long, others use it as 4 bytes long, which is the same as an int.

This is because,  int is defined as being 2 bytes or 4 bytes long, with the majority of compilers before the year 2000, choosing 2 bytes which ranged between [-32,768, 32,767] and was fine too be used as a pointer on 16 bit machines. So at 4 bytes, long was much larger than int at that time.

Now, 32 bit machines are the norm, so almost all compilers define int and being 4 bytes instead of 2, which makes them identical to the original standard setting for long.

In the C standard long is defined as being "at least" 4 bytes long. Some compilers therefore use the 4 byte version that as they always had. Others have "doubled" it size to 8 bytes []. The compiler I use uses the 8 bytes definition, so I didn't notice the problem.

Consequently, you can use "long long" instead of "long". If you use "long long" you cannot  use the -ansi flag with gcc. The makefile provide in the template does not use -ansi.

Mark Wineberg

PS All this will probably change again with the increased use of 64 bit machines. This is what is causing the problem with the new Mac OS Mohave, which demands 64 bit compliance, and is causing many programs to not run on the new operating system.

**“What you mean by implementing the list of commands? In my programs I parse through the input files and if the commands are there I call my functions”**

That is implementing the commands.

**“Are we required to make/ use the command.h and command.c files given?”**

Only if you \*do not want\* the bonus +5 marks. If you use the template, or template structure, you do not get the bonus. If you use your own, you do.

Mark Wineberg

**“In the recent Q and A (#6) you mentioned that we cannot use “internal” functions, nor can we use functions that take a node as a parameter or return a node. I was wondering what is meant by internal functions, and if this means that we can only use the functions which are required to be made in the assignment description.**

**For example, in my push() function, I have a function with the signature “void swap(Node \*, Node \*)”. Would the swap() function be not allowed because it takes nodes as parameters?”**

Your function swap is not a function the users of the ADT know about, or should know about.

By definition, this is an *internal function*, as opposed to the functions that the users are aware of, which are called *public functions*.

As it is not public, swap's signature should not placed in the sort\_list.h header file, which would make it accessible to users in their main()s or other functions. Consequently they could write their own swap functions and not be in conflict with yours.

If you did put swap in your sort\_list.h, marks would not be taken off, but consider moving it out.

Mark Wineberg

**“I was wondering how this "good effort" thing works. Is it like you complete Q1 and then any other question and you can still receive a high grade lets say an 80 on the entire assignment or do you have to complete all of them to get a grade above an 80. Another way to understand would be is the "good effort" just to receive the different methods of mark distribution (if that makes any sense).”**

Your grade for A4 is your grade for A4 based on the marking guideline posted.

You cannot receive a high grade on A4 if you only do Q1a and just try another question.  Nor would you get a high grade if you get perfect, say on Q1a and Q2. You would then get 50 + 25 = 75 out of 150.

Whether you pass Q1a and try one other question means that your A4 (calculated as above) could be weighed less than it would according to the new course outline, and not hurt you as much.

If you do extremely well on A4 by doing all of the questions and making almost no mistakes, then your A4 could be worth more than it would according to the new course outline, which could potentially raise your grade by a lot (or keep your grade high).

Mark Wineberg

**“For the function map\_2\_array, if the sizes of the lists don’t match? Should we return NULL?”**

Yes.

Mark Wineberg

**“In A4Q1b you ask us to return NULL in the sum\_of\_sq\_diff function but the function is meant to return something of type value\_type. value\_type in this case is an int where NULL is a pointer so how should I go about doing this?”**

The NULL is from an old design that I had changed; but missed this. Based on the new design, I would have it return FAILURE or SUCCESS and pass the value out using a parameter.

I have changed the assignment to have either return values acceptable, although FAILURE or SUCCESS is the better approach.

Mark Wineberg

**“I am still a bit confused on “private” functions in ADT’s and in what files I should place them in. As of now I have been using #include’s at the top of my command function’s files.**

**As an example, if my push function implements a private function called add\_front, the same way as in the lecture notes, I have been writing my code the following way:**

#include “add\_front.h”

int push(Sorted\_List \*myList, value\_type myVal, key\_type myKey){

          Node \*node = add\_front(&(myList -> head), myVal, myKey);

}

**Looking at the template for q1a, should my private functions (ex. add\_front) be included in the commands.c file, in a new file for private functions, or should it be left as is?”**

The add\_front signature (and hence in your case the add\_front.h) should not be added to command.h.

A function is "public" if its .o file is accessible to the user to link, or if the function is in a .a and its signature is known to the user to write it at the top of the file.

If you add the signature to the  .h that is naturally being included, as is command.h, along with command.c/command.o   then it is a public function, not a private function (aka an implementation function, or hidden function).

So basically just do as you have been doing. It is correct.

Mark Wineberg

PS  Having one file per function is not usually the way experienced programmers structure code (unless it is a \*very large\* function that can't naturally be broken up into smaller functions).

Usually you would group functions that have a similar purpose into a single file. If the file gets too large, then this implies that you have defined "similar purpose" too broadly. This allows for better readability of the file without having to jump from file to file to understand the flow of the program.

Nathan's file "design" is a good design given the sizes of the functions, which he has the experience to properly estimate as he is a third/fourth year student and has written a lot of code at this point.