# CS278 Assignment 15: Ch. 10 Advanced Macros

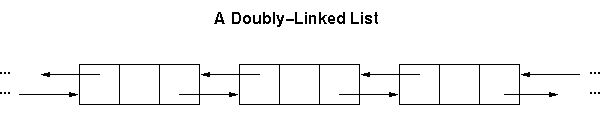
## NAME: Gabe Conlon DUE: Nov 15 Received: .

**GRADE:**

|  |  |  |
| --- | --- | --- |
| **CATEGORY** | **POINTS** |  |
| EX15\_01 |  | 50 |
| EX15\_02 |  | 50 |
|  |  |  |
| **TOTAL** |  | 100 |

## EXERCISES:

**EX15\_01 –** Write a macro that will build a doubly linked list. A doubly linked list is like a regular linked list, except each node has three parts, the first part points back at the node in front of it, the middle part is the data section, and the last part points at the next node in the linked list. Note that the very first node will not be able to point to the node in front of it, so that node’s first part should be null. The macro should use the REPEAT or WHILE functionality in order to build the linked list. The macro should be passed a number that will set up how many links should be set up in the doubly linked list. The data portion of the linked list nodes should be loaded with the counting numbers. Note that this assignment is very much like the example of a singly linked list in the text book in section 10.4.5 pp 412-414. There they show you how to have the macro build the list and how to later write out the values in the data cells of the linked list. Your program should operate in almost the same way, building the linked list and then during run time travelling down the list and outputing the contents of the data portion of the nodes until you get to the ending null pointer that signifies the end of the list (the book also has good examples for doing this). Your doubly linked list should end with a null pointer in the same way the book’s example does.



*Note: The link that points at the node in front of it really points to the beginning of the node, not the end of it! Another drawing showing the beginning and end:*

head

|

v

------------- ------------- -------------

| | | | <---+-- | | | <---+-- | | |

| 0 | a | --+---> | | b | --+---> | | c | 0 |

------------- ------------- -------------

**EX15\_02 –** Write a macro that finds the largest value in an array of cells and outputs that largest value. Note that the data in the array should not be sorted, so it is not always just the last value. The array can be any length and the size of the cells can be “BYTE” (bytes), “WORD” (word), “DWORD” (doubleword). There is an excellent example in the book that goes through the details of how to build a macro that can accept the cell size as an input parameter (Section 10.3.6 Example: Summing a Matrix Row pp 401-404). The call to the macro might look like this:

**FindMax MACRO Array2Search, Arraylength, CellSize**

INPUTS:

Array2Search – Name of array to search, call the macro at least three times with different array names and with different cell sizes (see below). This will mean that you will need to have at least three arrays set up. They need not be long, but they should be long enough to adequately test your macro. Note that if no array name is provided, the macro should echo a warning and exit early without expanding the code. (Hint: use the conditional assembly directives and the EXITM macro keyword as discussed in section 10.3)

Arraylength – the number of cells in the array – if no length is provided,the macro should echo a warning and exit early without expanding the code

CellSize – either “BYTE”, “WORD”, or “DWORD” If no size is provided, it should default to “BYTE” (hint: check section 10.3.2 Default Argument Initializers).

Note that there are implications for your movement through the array based on how big each cell is. The text book has an excellent example of how to work through this (the shift SHR is based on the TYPE CellSize / 2).

OUTPUTS:

The largest value in the array is output to the console. Use a nested macro call to a macro that outputs the value with a short message. If there is more than one cell with the largest value, the behavior is the same, just put out “The largest value found was” 189. (for example)

*Note that I have only assigned 2 programs again for this assignment. This is so you will have time to work on your projects!*