

Practical 8

This prac consists of 2 sections. The main one is interfacing with the LCD module, for 6 marks. The other is a small programming task for 4 marks.

LCD:

Collect a DIP module and veroboard from the white lab.

Build the circuit described in class such that the LCD can be controlled via the DIP module.

Recommendations:

- test your LCD screen before starting. Download and run the demo.elf program from Vula to test the screen.
- be very careful when powering the LCD screen. Connecting it incorrectly will almost certainly kill it!
- cut the tracks on the veroboard before soldering the DIP module in.
- cut all 10 wires to the same length and strip both ends of the wire before soldering them. The end going into the breadboard should be stripped a lot more than the end being soldered into the veroboard.
- do a continuity test between each adjacent track of the veroboard to ensure no bridges.
- after connecting the DIP module, measure each pin of the LCD screen to check that it goes high and low as expected when toggling its switch.
- if you want the LCD to return to its default state, remove power for 10 seconds and then power it up again.
- keep E idling high. When you want to send a byte to the LCD, pull it low and then bring it high straight after.
- remember that because the switch pulls the line LOW, if the switch is ON the line is low.

Initiliasie the LCD by sending in the following commands:

First:

Bit Number	7	6	5	4	3	2	1	0
Value:	0	0	1	1	1	1	x	x

That should result in the dark blocks going away, leaving the whole screen blank

Then:

Bit Number	7	6	5	4	3	2	1	0
Value:	0	0	0	0	1	1	1	1

This should result in a blinking, underlined cursor appearing.

Now switch to character mode and write the initials of one of your group members at the start of the first line. Demonstrate this to a tutor. The tutor should indicate this on their mark sheet. Make sure that they do! This will earn you 4 marks.

Next, write the initials of the other group member at the END of the second line. Again, demonstrate this to a tutor and they should award you the other 2 marks for this.

Your screen should look like:

A	B														
														X	Y

Programming task

This is a small programming task to get you familiar with pointers and arrays. It's only ~10 lines of code you need to write.

Attached is a template you should use.

In the template, space for an array is allocated in memory and then each element set to a value. It needs to be done this yucky, crude way because of reasons.

The array is guaranteed to be sorted: smallest element to largest element.

Two pointers are defined. The objective is to get these pointers to point to the two elements corresponding to the closest adjacent pair of elements. Very similar to prac3 and prac4!

They are pointers. They must **not** hold the values of the elements themselves. They should point to the values which were set in the array.

All you need to do is to implement the function `find_closest_pair` and write the call to the function.

Once the function has executed, the two pointers should point to the elements of the closest adjacent pair.

Do not make any changes to main other than adding call to `find_closest_pair`.

The automarker will change the contents of the array prior to running `find_closest_pair`

Marking:

After `find_closest_pair` runs, if the pointers are pointing to the correct elements: 4 marks.

If your code generates any compile warnings, a mark will be capped at 50%. You must compile with the -Wall flag.