

## Practical 3

Please use the template provided.

### Part 1: (2)

Iterate through each byte in our data block (defined starting at DATA) and push each byte to the stack. Note: each byte must be pushed individually. This needs to be completed when the program hits the instruction labeled `stack_push_done`. At this point the automaker will halt execution, verify the contents of the stack and modify the data on the stack.

### Part 2: (2)

Iterate through each element on the stack, and find the one element which is the largest and one which is the smallest. We will call these elements A and B respectively. Toggle between displaying A and B on the LEDs. The value displayed should change every 0.6 seconds (~2%). I.e.

Display A  
Delay for 0.6 seconds  
Display B  
Delay for 0.6 seconds  
Loop

### Part 3: (1)

If SW0 is pressed, display A-B constantly on the LEDs. If SW1 is pressed display A+B constantly on the LEDs.

**Part 4: (2)** If SW2 is held down, rather than toggle between A and B, display a value proportional to the value of POT1.

When the pot is rotated fully anticlockwise, the LEDs should display 0.

When the pot is rotated fully clockwise, the LEDs should display 0xFF.

It must be linear in between.

If SW2 is released, return to toggling between A and B.

### Bonus: (1)

When SW3 is held, display the temperature as measured by the TC74A0 on LEDs. The value on the LEDs should correspond to the temperature in degrees celsius.

Available marks: 8 Marked out of: 7

**Hints:**

Part 1: Be careful how to treat the end pointer. You must make sure all values are copied. You can examine the contents of the stack with the gdb 'x' command.

Part 2: You'll probably need to keep track of two state variables: Best A and best B found so far. You'll need to keep registers dedicated to this.

Parts 3 and 4: Only one switch will be held down at any one time.

Do not hardcode the functionality.