Practical 3

Attached is a template which you should use.

Part 1: (2)
Iterate through each byte in our data block (defined starting at DATA_START) 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 two elements which are numerically closest to each other. We will call these elements A and B.
Note: when iterating through the elements on the stack, it is guaranteed that the elements will go from smallest at the top of the stack to largest at the bottom. Ie: The data at address n will be smaller than the data at n+1.

Part 3: (1)
Toggle between displaying A and B on the LEDs.
The value displayed should change every 1 second.

Part 4: (2)
While SW0 is held down, rather than toggle between A and B, display a value proportional to the value of POT0.
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 SW0 is released, return to toggling between A and B.

Bonus: (1)
If SW1 is held down, the rate of toggling should be controlled by data arriving on the RS-232 port. ASCII values between ‘0’ and ‘9’ will be sent. If ‘9’ is received, the toggle rate should be 1 second. If ‘0’ is received the toggle rate should be 0.2 seconds. Linear in between. Data won’t be sent faster than one byte every two seconds.

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 three state variables:

- Best A and best B found so far, where ‘best’ is defined as the closest pair. That’s two variables.
- value of the difference between best A and B

You’ll need to keep registers dedicated to these purposes
You could probably do away with storing the difference and rather re-calculate it each time if you prefer.

Part 3:
An easy way to toggle is to check: is the current value on the LEDs == A? If so, display B, if not display A.