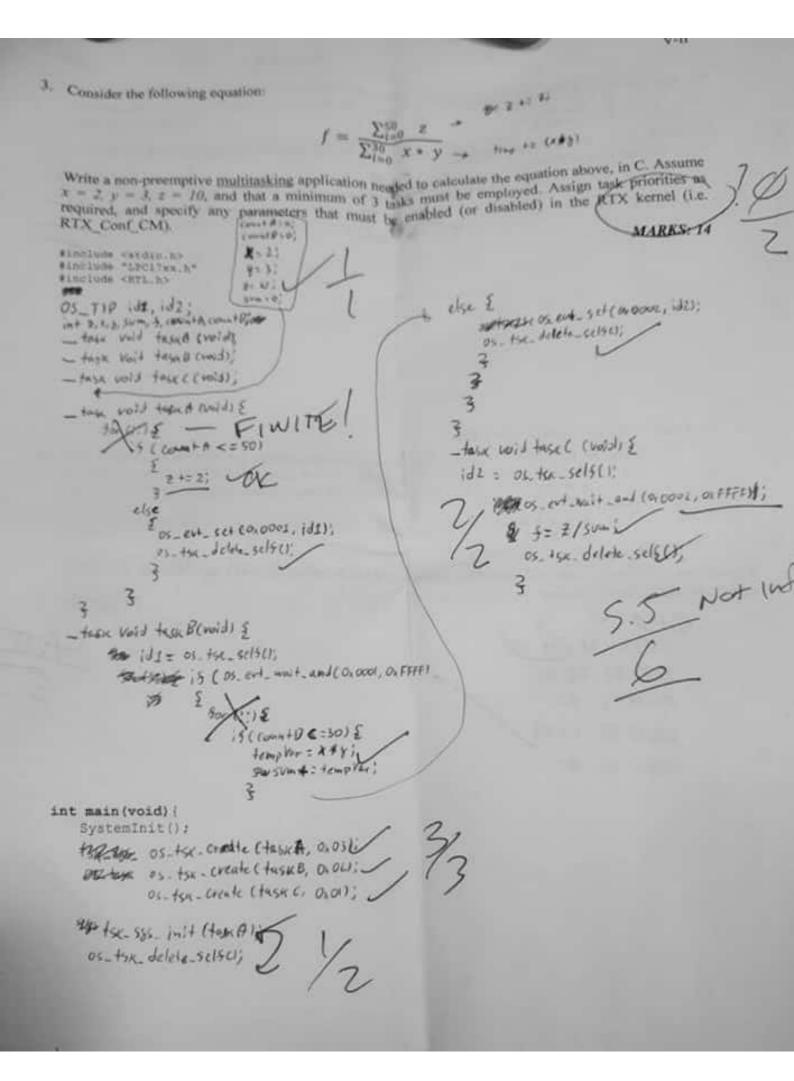


Indica	te (in the space provided) whether the following statements are TRUE or FALSE. To obtain full
marks	te (in the space provided) whether the following statements are to the for each question, include SHORT comments in support of your answer.  MARKS: 12 (2 each)
a)	The tail chaining feature in the ARM Cortex-M3 is used to improve contexts witching.  TRUE v or FALSE ?  And of the NVIC.
	The instruction BL <addr> moves the PC address found in R14 to the Link Register R15, and then branches to the location specified in <addr>.  TRUE or FALSE x_?  PC is R15 Link Rosister is R19</addr></addr>
-	3+ also supports tail-draining and dynamic to terupts
d) AR TR	M's R-Series line consist of CPUs dedicated to Real-Time applications.  UE **_ or FALSE?  A - Series are far High-personneys, R - Series for Real - Time, and M-series.
	ISC-type instruction set may not be implemented as a Von Neumann model.  IE or FALSE \( \frac{1}{2} \)  CLSC-type instructions are are left then fore can be
-	Implemented as a / Von Neumann model.
	estruction ADDS.W R2, #1 belongs to ARM Thumb2's 16-bit instruction set.
	W is for word which is 32.835



4. (a) Consider the following C code snippet:

$$x1 = x2 + (x3/2) / x5 = x1 + (x4*8) / x5$$

Convert the code above to ARMv7 assembly. Assume that the compiler flag -O3 has been set, and that

there is no dead code elimination.

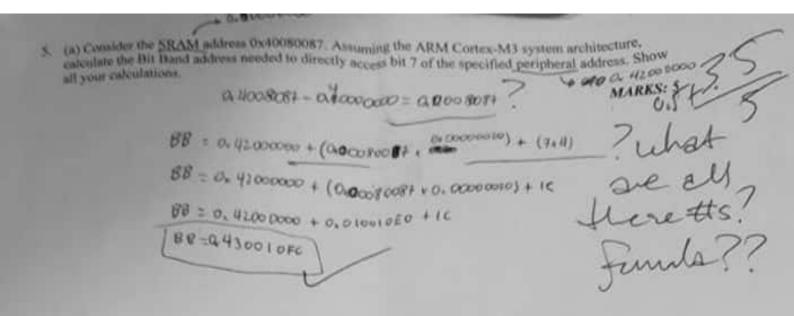
MARKS: 5

4. (b) Consider the following C code snippet:

Convert the code above to ARMv7 assembly. Assume that the compiler flag -O3 has been set, and that

ITTEE GE (R1 . 82 SUBGE R1 R2 R3 MOVERS #0 ADDLT RI RS RA MOVET R3, #5

MARKS: 5



5. (b) Integrate the bit banding address calculated in (a) into the C code below. The program should setup the address as a variable to be used in main. The main function should set the bit band variable to 1, delay for 50 ticks using os\_delay(), and then clear the same variable (i.e. to 0).

#Include \*stdio.h>
#Include \*LPC17xx.h"

SET R'= 0,43pointc

# Alefive

Lint R1' = R;

A

int main(void) (

R1 = 1;

os\_delay (50);

R1 = 0;