NOTE: You should submit the scan/image of your answer script on the moodle submission link within the give time-limit. There will be no time extension for submission, and your paper will not be accepted in any other mode.

Course Learning Outcomes (CLOs)

1. Understand the specific requirements and problems arising in embedded system application design and development.
2. Describe and explain fundamental concepts related to embedded system architecture and different components within the system.
3. Describe and explain the hardware-software interfaces, the memory architecture, and communication modes between components in embedded systems.
4. Produce efficient real-time designs using latest technology of NXP-LPC microcontrollers based on ARM cortex-M3 core.
5. Use various subsystems of a microcontroller in practical applications.
6. Implement and debug embedded systems software for simple applications combining assembly and C language programming.

Question	1	2	3	4	5	6	7	8	TOTAL
CLOs	2,6	1,4	3,5	3,5	3,5	1,2,4	4,5,6	4,5,6	
Value	10	10	10	15	15	10	15	15	100
Result									

1. Define the following C function using assembly instructions by following the arm architecture procedure call standard (AAPCS). (10)

```
int calc(int*a, int b)
{
   int c = 0;
   for (int i = b; i > 0; i--)
   {
        if(a[i-1] > 0)
             c = c + a[i-1];
        else
             c = c - a[i-1];
   }
   return c;
}
```

2. Design a finite state machine showing the operation of a simple stop watch with three input buttons: start, stop, reset.
   - <reset>: will clear the value stored in the counter.
   - <start>: will start counting – updating the counter.
   - <stop>: will stop the counting. The value in counter is not updated further.
   - The counter value is the output of the machine. (10)

3. Write a C function `Init` that configures the pins of a LPC1769 MCU as following: Port3 - pin 13 and 14 as outputs, and Port2 - pins 10 and 12 for reading inputs. Derive a high output at the port3-pin 13 and low output at the port3-pin 14 in initialization. The function returns void and does not require any input. (10)

4. Consider you are interfacing a Common anode seven segment display with LPC1769 MCU to show even numbers between 0 and 9 i.e. (0, 2, 4, 6, 8).
   - Port 1: pin 4 to pin 11 are used to connect to the LED segments (a to g, and dp) in given order.
   - Port 1: pin 0 and 1 are used for driving the common anode terminals of the display.

   Complete the C program template for this task given below. The template consists of the data structure storing active segment's information for counting from 0 to 15 based on MCU pin selection (`display_code`) given above. Use simple loop for incorporating delay between two display values. Demonstrate all steps: Pin function configuration, mode selection (output and input), and writing/reading of pin values correctly. (15)

```
#include <LPC17xx.h>
unsigned int display_code[16]={0x000003f0,0x00000060,0x000005b0,0x000004f0,
                               0x00000660,0x000006d0,0x000007d0,0x00000070,
                               0x000007f0,0x000006f0,0x00000770,0x000007c0,
                               0x00000390,0x000005e0,0x00000790,0x00000710};

int main(void)
{
```

5. Consider you are interfacing a switch and two LEDs (L1 and L2) with LPC1769 MCU.
   - When the switch is ON (logic HIGH), L1 is ON, and L2 is OFF.
   - When the switch is OFF (logic LOW), L1 is OFF, and L2 is ON.
   - Switch is interfaced with port 2 – pin 0, LED L1 with port 2 – pin 4, and LED L2 with port 2 – pin 5.
   - Initially L1 is ON, and L2 is OFF.

   Complete the C program template for this task given below. Demonstrate all steps: Pin function configuration, mode selection (output and input), and writing/reading of pin values correctly. (15)

```
#include <LPC17xx.h>
int main(void)
{
```

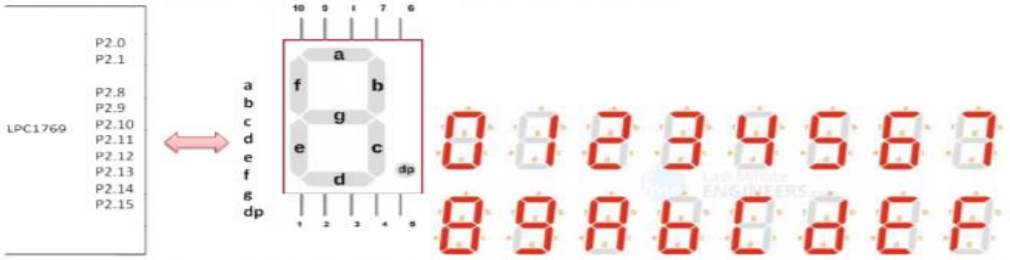
6. Consider you are using PLL0 module of LPC1769 for clock generation. The PLL0 is using the main oscillator (crystal) available in LPC1769 MCU as source. You want to generate 240MHz clock from the PLL (`pll0clk`). What are possible values for PLL0 N-Divider's factor register and PLL0 internal clock divider's factor register which could generate this clock frequency? (10)

7. Define a C function named `initTimer` which initializes Timer1 (`TIM1`) module in LPC1769 module. Assume the CPU clock is set at 50MHz. The peripheral clock for the Timer1 should be set at half of the clock frequency (PCLK = CCLK/2). The timer resolution is 0.001 second. Set the Timer1 to timer mode and set the appropriate prescalar value. You should also reset the timer in initialization. The function does not take any input, and returns `void`. (15)

8. Consider you are interfacing a Common cathode seven segment display with LPC1769 MCU to show even numbers between 0 and 9 i.e. (0, 2, 4, 6, 8).
   - Port 2: pin 8 to pin 15 are used to connect to the LED segments (a to g, and dp) in given order.
   - Port 2: pin 0 and 1 are used for driving the common cathode terminals of the display.

   Write the C program to complete this task. Use simple loop for incorporating delay between two display values. Demonstrate all steps: Pin function configuration, mode selection (output and input), and writing/reading of pin values correctly.

   The corresponding design is given below for your understanding.



   The table consisting of the information of active LED segments for counting from 0 to 15 is given below

Output (Hex)	Segments							Value on Display
	g	f	e	d	c	b	a	
0x3F	0	1	1	1	1	1	1	0
0x06	0	0	0	0	1	1	0	1
0x5B	1	0	1	1	0	1	1	2
0x4F	1	0	0	1	1	1	1	3
0x66	1	1	0	0	1	1	0	4
0x6D	1	1	0	1	1	0	1	5
0x7D	1	1	1	1	1	0	1	6
0x07	0	0	0	0	1	1	1	7
0x7F	1	1	1	1	1	1	1	8
0x67	1	1	0	0	1	1	1	9
0x77	1	1	1	0	1	1	1	A
0x7C	1	1	1	1	1	0	0	b
0x39	0	1	1	1	0	0	1	C
0x5E	1	0	1	1	1	1	0	d
0x79	1	1	1	1	0	0	1	E
0x71	1	1	1	0	0	0	1	F

(15)