AIN SHAMS UNIVERSITY
FACULTY OF ENGINEERING

Juba

Computer and Systems Engineering Department
Junior Electrical Engineering (Computer & Electronics & Power)

Spring 2021	**Course Code: *CSE 211***	***Time allowed: 2 Hrs.***

INTRODUCTION TO EMBEDDED SYSTEMS

The Exam Consists of **60** Questions in **12** Pages. ***Maximum Marks: 60 Marks*** **1 / 12**

تعليمات هامة

- حيازة التيلفون المحمول مفتوحا داخل لجنة الأمتحان يعتبر حالة غش تستوجب العقاب وإذا كان ضرورى الدخول بالمحمول فيوضع مغلق فى الحقائب.
- لا يسمح بدخول سماعة الأذن أو البلوتوث.
- لايسمح بدخول أي كتب أو ملازم أو أوراق داخل اللجنة والمخالفة تعتبر حالة غش.

Power: mark Exam Model as A
Comp & Comm: mark Exam Model as B

For each of the following 60 multiple choice questions (MCQs), select ONLY the ONE correct answer. Mark your choice on the answer bubble sheet. ..... [The 60 MCQs are equal in weight]

1. How many general-purpose registers do ARM Cortex-M processors have?

A) 10	B) 11	C) 13	D) 15

2. Which register is used to point to the next instruction to be fetched in ARM Cortex-M processors?

A) R13	B) R14	C) R15	D) R16

3. Which register is used to store the return address in ARM Cortex-M processors?

A) R13	B) R14	C) R15	D) R16

4. Which register is the stack pointer in ARM Cortex-M processors?

A) R13	B) R14	C) R15	D) R16

5. Which PSR flag in ARM Cortex-M processors indicates a negative result?

A) Z flag	B) N flag	C) C flag	D) V flag

6. Which PSR flag in ARM Cortex-M processors indicates an unsigned overflow?

A) Z flag	B) N flag	C) C flag	D) V flag

7. Which PSR flag in ARM Cortex-M processors indicates a signed overflow?

A) Z flag	B) N flag	C) C flag	D) V flag

8. Which PSR flag in ARM Cortex-M processors indicates a zero result?

A) Z flag	B) N flag	C) C flag	D) V flag

9. What is the size of the Flash ROM in the TM4C microcontroller?

A) 32 KB	B) 64 KB	C) 128 KB	D) 256 KB

10. Which bus is connected to the Data RAM in ARM Cortex-M processors?

A) Advanced High-perf Bus (AHB)	B) DCode bus	C) ICode bus	D) System bus

11. Which bus(s) is(are) connected to the Instructions Flash ROM in ARM Cortex-M processors?

A) DCode bus	B) ICode bus	C) System bus	D) Answers (A) and (B)

12. A serial communication channel that does not have a clock signal is said to be

A) Asynchronous	B) Parallel	C) Synchronous	D) UART

13. The letter A in UART stands for

A) Accumulator	B) ASCII	C) Asymmetric	D) Asynchronous

PROG2 Q25—Q32	
1	void UART_Init(void); void UART_OutChar(char data){
2	while((UART0_FR_R&0x0020) != 0);

3	UART0_DR_R = data;
	}
	int main(void){
	UART_Init();
	for(;;){

4	UART_OutChar(<C1>);
5	} }
6	void UART_Init(void){ SYSCTL_RCGCUART_R |= 0x00000001; SYSCTL_RCGCGPIO_R |= 0x00000001; UART0_CTL_R &=!0x00000001; UART0_IBRD_R=43; UART0_FBRD_R=26;
7	UART0_LCRH_R=0x00000070;
8	UART0_CTL_R=0x00000301;
9	GPIO_PORTA_AFSEL_R |= 0x03;
10	GPIO_PORTA_PCTL_R = (GPIO_PORTA_PCTL_R &0xFFFFFF00)+0X0000011;
11	GPIO_PORTA_DEN_R |= 0x03;
12	GPIO_PORTA_AMSEL_R &= ~0x03;
13	}

25. In PROG2, if bus clock frequency is 80 MHz, the approximate baud rate, in bits per second, is

A) 4,600	B) 9,600	C) 19,200	D) 115,200

→ $DivFrac = FBRD * 64 + 0.5$

∴ $FBRD = \frac{DivFrac - 0.5}{64} = \frac{26 - 0.5}{64} = 0.3984$

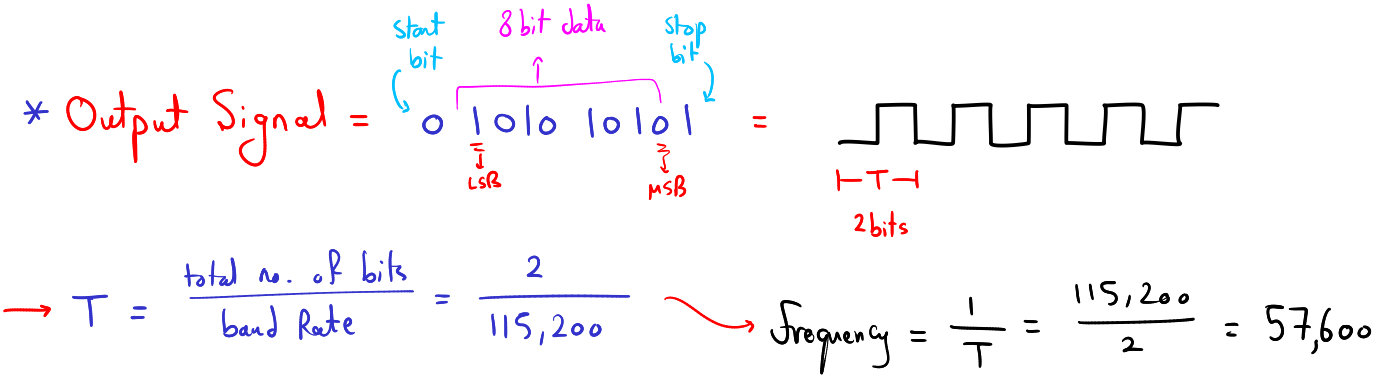
→ $BRD = IBRD + FBRD = 43 + 0.3984 = 43.3984$

→ $baud\ Rate = \frac{Clock\ Freq.}{16 * BRD} = \frac{80 \times 10^6}{16 * 43.3984} \simeq 115,200$

MSB 0101 0101 LSB

26. In PROG2, if <C1> in line 4 is set to 0x55 and noting that the UART is configured to handle 8-bit data with no parity and one stop bit, the approximate frequency, in Hz, of the signal generated on serial output is

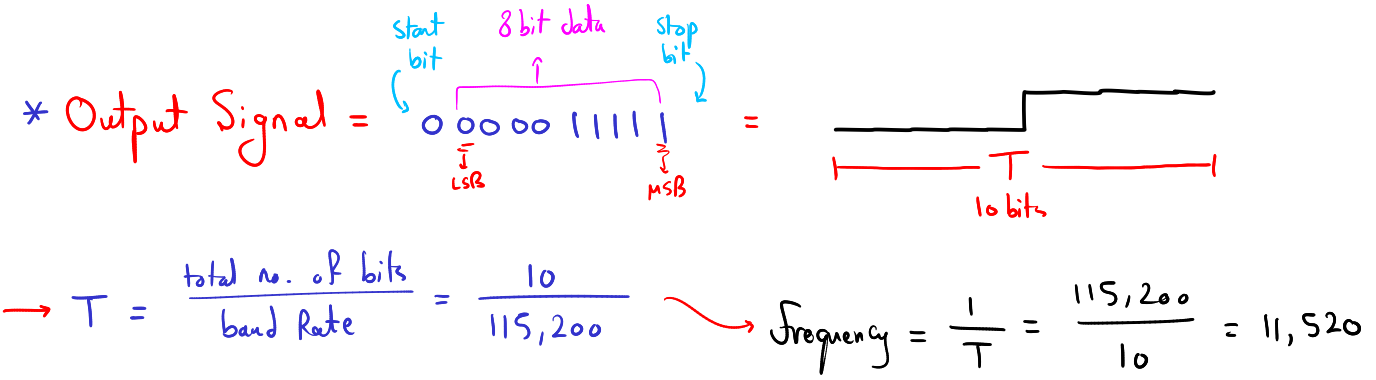
A) 9,600	B) 28,800	C) 57,600	D) 115,200



1111 0000

27. In PROG2, if <C1> in line 4 is set to 0xF0 and noting that the UART is configured to handle 8-bit data with no parity and one stop bit, the approximate frequency, in Hz, of the signal generated on serial output is

A) 57,600	B) 11,520	C) 115,200	D) 1,152,000



28. In PROG2, the functionality of line 2 is to wait until

A) a new input is available	B) receive buffer is not empty	C) RXFE is 0	D) TXFF is 0

```
2   while((UART0_FR_R&0x0020) != 0);
```

29. In PROG2, the functionality of line 2 can be described as

A) busy-waiting	B) echoing	C) inspection	D) reviewing

30. In PROG2, the UART is configured to use one stop bit in → LCRH

A) Line 7	B) Line 8	C) Line 9	D) Line 10

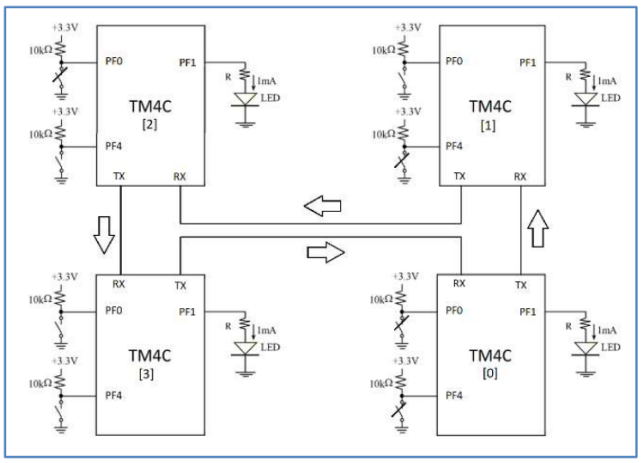
31. In PROG2, the UART is enabled in → CTL

A) Line 7	B) Line 8	C) Line 9	D) Line 10

32. In PROG2, the UART is configured to have no parity in → LCRH

A) Line 7	B) Line 8	C) Line 9	D) Line 10

Q39—Q44: Four TM4C microcontrollers are connected in a ring using UART, as shown in the figure to the right. The transmitter line of one microcontroller is connected to the receiver line of its neighbor to form a ring. Each microcontroller defines a unique address for itself using two dip switches connected to PF0 (least bit) and PF4. Exchanged messages consist of 3 characters; first character defines the sender address; second one defines the receiver address; and the third defines the receiver LED state to set to. As an example, if microcontroller 0 wants to switch on the LED on microcontroller 2, the message will be "021" and to switch it off, the message should be "020". When receiving a message, a microcontroller checks the second character, if it matches its address, then the microcontroller should apply the received setting to its LED; otherwise, the microcontroller forwards the message to its neighbor. If a message comes back to its sender that means that the receiver address does not match any microcontroller on the ring and that message should be dropped. PROG4 (on the next page) is running on each microcontroller. In PROG4, each microcontroller tries to switch its LED ON through the ring connection. Some statements in PROG4 are not in the right order.



39. In PROG4, line 2, expression <E1> (used to get the microcontroller own address should be

A) A0+A1	B) A0*2+A1	C) A0*2+A1*4	D) A0+A1*2

```
   A0=Get_PIN(GPIO_PF0_M);
   A1=Get_PIN(GPIO_PF4_M);
2  myAddress=<E1>;
```

least bit

∴ Address = $A_0 * 2^0 + A_1 * 2^1$

$= A_0 + A_1 * 2$

40. In PROG4, lines 3 and 6, <X> should be

A) 0	B) '0'	C) 1	D) '1'

```
3  UART_OutChar(myAdd+<X>);
   UART_OutChar(myAdd+<X>);
   UART_OutChar(ON+<X>);
```

→ to convert an integer no. to it's equivalent character in Ascii code ⇒ add it to '0'

41. In PROG4, line 5, assuming that <X> is set as in the previous question, condition <C1> should be

A) RecAdd==myAdd+<X>	B) RecAdd!=myAdd+<X>	C) TranAdd==myAdd+<X>	D) TranAdd!=myAdd+<X>

```
5  if (<C1>){
6     if (LED_state==ON+<X>) Set_PIN(GPIO_PF1_M, ON);
      else Set_PIN(GPIO_PF1_M, OFF);
```

→ check if second character = myAddress

↳ Receiver address

42. In PROG4, line 8, assuming that <X> is set as in the previous question, condition <C2> should be

A) RecAdd==myAdd+<X>	B) RecAdd!=myAdd+<X>	C) TranAdd==myAdd+<X>	D) TranAdd!=myAdd+<X>

it means <C1> haven't been satisfied

```
7  }else{
8     if (<C2>){
9        UART_OutChar(TranAdd);
10       UART_OutChar(LED_state);
11       UART_OutChar(RecAdd);
```

∴ in case of second character not equal to my address

Check for first character (transmitter address)

if it's equal to my address ⇒ drop data "skip"

if it's not equal to my address ⇒ forward to neighbour "print"

43. In PROG4, the functionality of line 14 is to wait until ;btw choices a,b and c have the same meaning

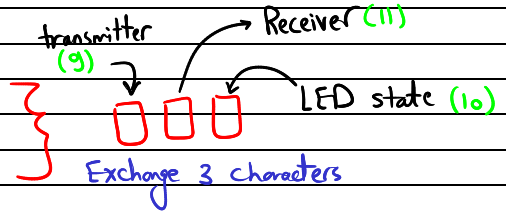
A) a new input is available	B) receive buffer is not empty	C) RXFE is 0	D) TXFF is 0

44. In PROG4, the correct order for lines 9—11 should be

A) 11-10-9	B) 10-11-9	C) 9-11-10	D) 10-9-11

PROG4 Q39—Q44

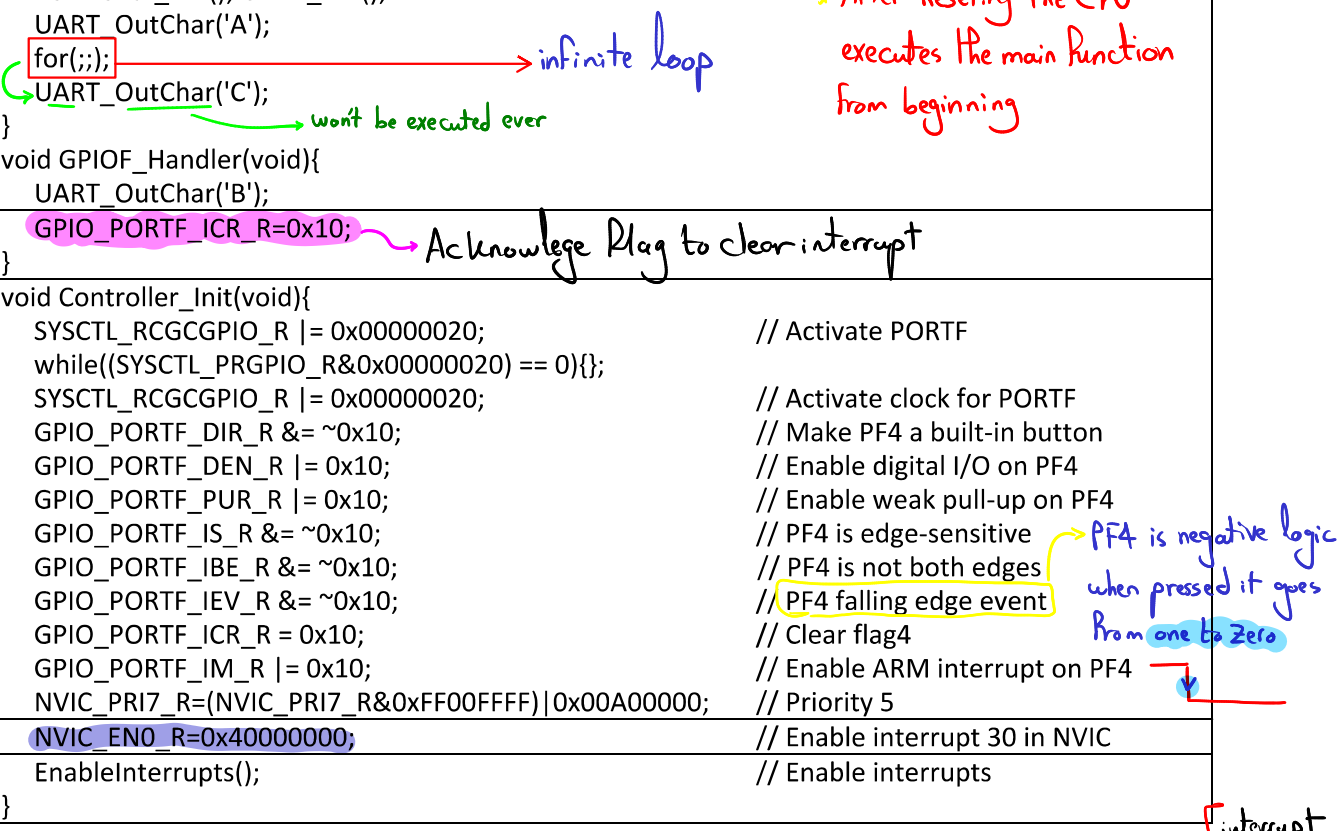
```
1  #define GPIO_PF0_M              0x01U     // PF0 is connected to a switch
   #define GPIO_PF4_M              0x10U     // PF4 is connected to a switch
   #define GPIO_PF1_M              0x02U     // PF1 is connected to a LED
   #define ON                      1
   #define OFF                     0
   void Controller_Init(void);
   unsigned char Get_PIN(unsigned char PIN);
   void Set_PIN(unsigned char PIN, unsigned char val);
   void UART_Init(void);
   void UART_OutChar(char data);
   char UART_InChar(void);
   int main(void){
      unsigned char TranAdd, RecAdd, LED_state;
      unsigned char A0, A1, myAdd;
      Controller_Init();
      UART_Init();
      A0=Get_PIN(GPIO_PF0_M);
      A1=Get_PIN(GPIO_PF4_M);
2     myAddress=<E1>;
3     UART_OutChar(myAdd+<X>);
      UART_OutChar(myAdd+<X>);
      UART_OutChar(ON+<X>);
4     for(;;){
         TranAdd=UART_InChar();
         RecAdd=UART_InChar();
         LED_state=UART_InChar();
5        if (<C1>){
6           if (LED_state==ON+<X>) Set_PIN(GPIO_PF1_M, ON);
            else Set_PIN(GPIO_PF1_M, OFF);
7        } else{
8           if (<C2>){
9              UART_OutChar(TranAdd);
10             UART_OutChar(LED_state);
11             UART_OutChar(RecAdd);
12          }
         }
      }
   }
13 char UART_InChar(void){
14    while((UART0_FR_R&0x0010) != 0);
15    return((char)(UART0_DR_R&0xFF));
   }
16 void UART_OutChar(char data){
      while((UART0_FR_R&0x0020) != 0);
      UART0_DR_R = data;
   }
```



Q45—Q50: A display is connected to a microcontroller running PROG8 to display all UART output.

PROG8: Q45—Q50 — سؤال مخصص لطلبة قسم حاسبات وطلبة قسم إتصالات فقط.

```
1  void Controller_Init(void);
   void UART_Init(void);                          // Configure UART0
   void UART_OutChar(char);                       // Send a char on UART0
   void EnableInterrupts(void);                   // Enable interrupts
   int main(void){
      Controller_Init(); UART_Init();
      UART_OutChar('A');
      for(;;);
      UART_OutChar('C');
   }
   void GPIOF_Handler(void){
      UART_OutChar('B');
2     GPIO_PORTF_ICR_R=0x10;
   }
3  void Controller_Init(void){
      SYSCTL_RCGCGPIO_R |= 0x00000020;            // Activate PORTF
      while((SYSCTL_PRGPIO_R&0x00000020) == 0){};
      SYSCTL_RCGCGPIO_R |= 0x00000020;            // Activate clock for PORTF
      GPIO_PORTF_DIR_R &= ~0x10;                  // Make PF4 a built-in button
      GPIO_PORTF_DEN_R |= 0x10;                   // Enable digital I/O on PF4
      GPIO_PORTF_PUR_R |= 0x10;                   // Enable weak pull-up on PF4
      GPIO_PORTF_IS_R &= ~0x10;                   // PF4 is edge-sensitive
      GPIO_PORTF_IBE_R &= ~0x10;                  // PF4 is not both edges
      GPIO_PORTF_IEV_R &= ~0x10;                  // PF4 falling edge event
      GPIO_PORTF_ICR_R = 0x10;                    // Clear flag4
      GPIO_PORTF_IM_R |= 0x10;                    // Enable ARM interrupt on PF4
      NVIC_PRI7_R=(NVIC_PRI7_R&0xFF00FFFF)|0x00A00000; // Priority 5
4     NVIC_EN0_R=0x40000000;                      // Enable interrupt 30 in NVIC
5     EnableInterrupts();                         // Enable interrupts
   }
```



45. In PROG8, the message displayed, if no action occurs on PF4 after resetting the microcontroller is

A) A	B) AC	C) ACB	D) ABC

46. In PROG8, the message displayed, if PF4 is changed from zero to one after resetting the microcontroller is

A) A	B) AB	C) ACB	D) ABC

47. In PROG8, the message displayed, if PF4 is changed from one to zero after resetting the microcontroller is

A) A	B) AB	C) ACB	D) ABC

48. In PROG8, the message displayed, if PF4 is changed from one to zero then from zero to one again after resetting the microcontroller is

A) A	B) AB	C) ACB	D) ABC

49. In PROG8, assume line 2 is removed, the message displayed, if PF4 is changed from one to zero then from zero to one again after resetting the microcontroller is

A) A	B) AB	C) ABB	D) ABBBBBBB ...to infinity

50. In PROG8, assume line 4 is removed, the message displayed, if PF4 is changed from one to zero then from zero to one again after resetting the microcontroller is

A) A	B) AB	C) ABB	D) ABBBBBBB ...to infinity

the interrupt won't be triggered as it haven't been enabled

Q49—Q55: PROG6 is used to drive a stepper motor with 200 steps/rotation. The clockwise rotation is achieved by applying the code 5,6,10,9. The motor is connected to PORTD. Motor rotation direction is selected based on a switch connect to bit 0 of PORTF (PF0), which when set to 1 enables a clockwise rotation.

PROG6 Q49—Q55		سؤال مخصص لطلبة قسم قوي فقط.
1	#define GPIO_PF0_M 0x01U #define STEPPER GPIO_PORTD_DATA_R	// PF0 is connected to direction switch

2	void Controller_Init(void);
	unsigned char SW_Input(void);
	void delay1ms(long delay);
		// Initialize PORTF and PORTD
		// Read the state of PF0 (switch)
		// Adding a delay (unit is 1 ms)

3	int main(void){
	unsigned char step_array[]={5,6,10,9};
	int current_index=0;
	Controller_Init();	

4	for(;;){ STEPPER=step_array[current_index];	
5	if (SW_Input()) <S1>	
6	else <S2>	
7	if(current_index<0) <S3>	
8	if(current_index> <N>) <S4>	
9	delay1ms(<X>);	
10	} }	

49. In PROG6, the motor step size is

A) 1.8°	B) 10°	C) 18°	D) 45°

→ Step Size = $\frac{360^\circ}{\text{no. of steps/revolution}} = \frac{360^\circ}{200} = 1.8^\circ$

50. In PROG6, line 5, statement <S1> should be

A) current_index -=1;	B) current_index -=2;	C) current_index+=1;	D) current_index +=2;

51. In PROG6, line 6, statement <S2> should be

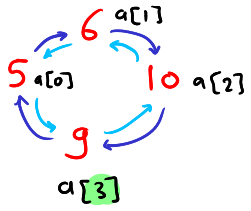
A) current_index -=1;	B) current_index -=2;	C) current_index+=1;	D) current_index +=2;

52. In PROG6, line 7, statement <S3> should be → 0, 1, 2, 3, 0, 1, 2, 3

A) current_index=0;	B) current_index=1;	C) current_index=2;	D) current_index=3;

53. In PROG6, line 8, <N> should be

A) 0	B) 1	C) 2	D) 3



* if PF0 is set to 1 → Clockwise
* if PF0 is set to 0 → Anti-clockwise

54. In PROG6, line 8, statement <S4> should be → 0, 1, 2, 3, 0, 1, 2, 3

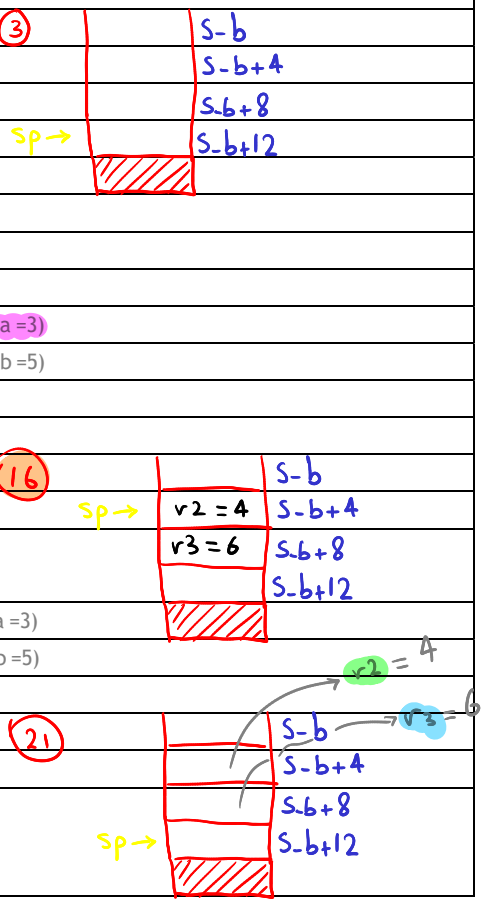
A) current_index=0;	B) current_index=1;	C) current_index=2;	D) current_index=3;

55. In PROG6, line 9, in order to have a motor speed of 12 RPM, <X> should be

A) 6	B) 12	C) 25	D) 50

→ delay in ms = $\frac{60 * 1000}{\text{no. of steps/rev} * \text{rpm}} = \frac{60 * 1000}{200 * 12} = 25$

PROG9 Q51—Q56				سؤال مخصص لطلبة قسم حاسبات وطلبة قسم إتصالات فقط.
1		AREA	WRITE_variables, DATA, READWRITE	
	a	space	4	
	b	space	4	
	s_size	equ	12	
	s_b	space	s_size	; Stack base address
		AREA	MYCODE, CODE, READONLY	
2		ldr	sp, =s_b	;initial sp with the array base address
3		add	sp, #s_size	
4		ldr	r0, =a	;r0 has the address of symbol a
5		mov	r4, #3	;r4 = 3
6		str	r4, [r0]	;store r4 to the address stored in r0 (a=3)
7		ldr	r1, =b	;r1 has the address of symbol b
8		mov	r4, #5	;r4 = 5
9		str	r4, [r1]	;store r4 to the address stored in r1 (b=5)
10		ldr	r2, [r0]	;load the content of address stored in r0 to r2 (r2=a =3)
11		ldr	r3, [r1]	;load the content of address stored in r1 to r3 (r3=b =5)
12		add	r2, #1	;r2 = r2+1 = 4
13		add	r3, #1	;r3 = r3+1 = 6
14		bl	func	
15		b	stop	
16	func	push	{r2-r3}	
17		ldr	r2, [r0]	;load the content of address stored in r0 to r2 (r2=a =3)
18		ldr	r3, [r1]	;load the content of address stored in r1 to r3 (r3=b =5)
19		str	r2, [r1]	;store r2 to the address stored in r1 (b=3)
20		str	r3, [r0]	;store r3 to the address stored in r0 (a=5)
21		pop	{r2-r3}	
22	stop	bx	lr	
		END		



51. In PROG9, the value of r2 at the end of the program is

A) 2	B) 3	C) 4	D) 5

52. In PROG9, the value of r3 at the end of the program is

A) 2	B) 4	C) 5	D) 6

53. In PROG9, the content of variable a in memory before calling function func is

A) 3	B) 4	C) 5	D) 6

54. In PROG9, the content of variable a in memory after calling function func is

A) 1	B) 3	C) 4	D) 5

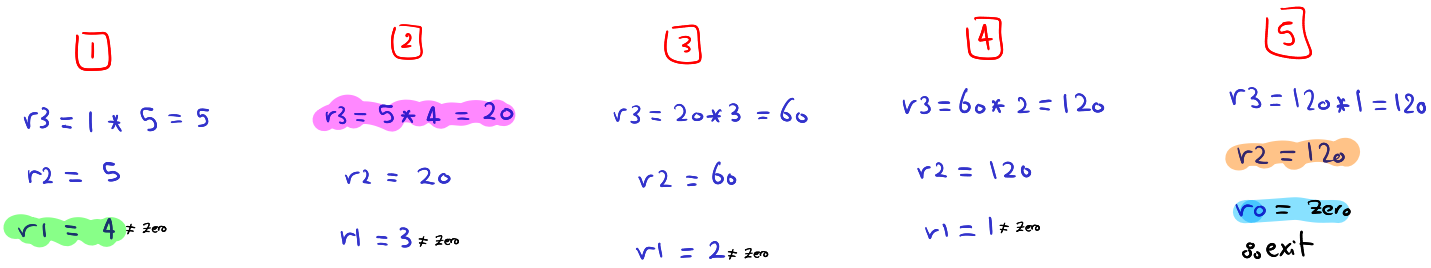
55. In PROG9, function func is used to

A) Swap the contents of registers r2 and r3	B) Swap the contents of registers r0 and r1
C) Swap the contents of variables *a* and *b* in memory	D) Swap the contents of registers r0 and r2

56. In PROG9, the value of SP after line 19 is

A) s_b-8	B) s_b+20	C) s_b+4	D) s_b+8

PROG10 Q57—Q60				سؤال مخصص لطلبة قسم حاسبات وطلبة قسم إتصالات فقط.
1		AREA	READ_variables, DATA, READONLY	
	A	DCD	5	
		AREA	WRITE_variables, DATA, READWRITE	
	Z	DCD	0	
		AREA	MYCODE, CODE, READONLY	
2		LDR	r0, =A	;r0 has the address of symbol A
3		LDR	r1, [r0]	;load the content of address stored in r0 to r1 (r1 = A = 5)
4		MOV	r2, #1	;r2 = 1
5		CMP	r1, #0	;r1 - 0 = 5 - 0 = 5 > 0
6		BLE	LOC2	;branch to LOC2 if less than or equal to zero
7	LOC1			
		MUL	r3, r2, r1	;r3 = r2 * r1
8		MOV	r2, r3	;r2 = r3
9		SUB	r1, r1, #1	;r1--
10		CMP	r1, #0	;r1 - 0
11		BLE	LOC2	;branch to LOC2 if less than or equal to zero
12		B	LOC1	;branch to LOC1
13	LOC2			
		LDR	r4, =Z	;r4 has the address of symbol Z
14		STR	r2, [r4]	;store r2 to the address stored in r4 (Z=)
		END		



57. In PROG10, the value of r1 after the first execution of line 9 is

A) 0	B) 4	C) 5	D) 24

58. In PROG10, the value of r3 after the second execution of line 7 is

A) 0	B) 1	C) 4	D) 20

59. In PROG10, the value of r1 after the execution of line 13 is

A) 0	B) 1	C) 4	D) 5

60. In PROG10, the value of r2 after the execution of line 14 is

A) 100	B) 120	C) 130	D) 150

END of Exam

Examination Committee
Dr. Ashraf Salem, Dr. M. Watheq El-Kharashi,
Dr. Mohamed Taher, and Dr. Ahmed M. Zaki.

Exam Date: 23rd of June, 2021