AIN SHAMS UNIVERSITY
FACULTY OF ENGINEERING

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 **<u>12</u>** Pages. ***Maximum Marks: 60 Marks*** **1 / 12**

<u>تعليمات هامة</u>

- حيازة التيليفون 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 <u>Exam Model as A</u>
Comp & Comm: mark <u>Exam Model as B</u>

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

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)

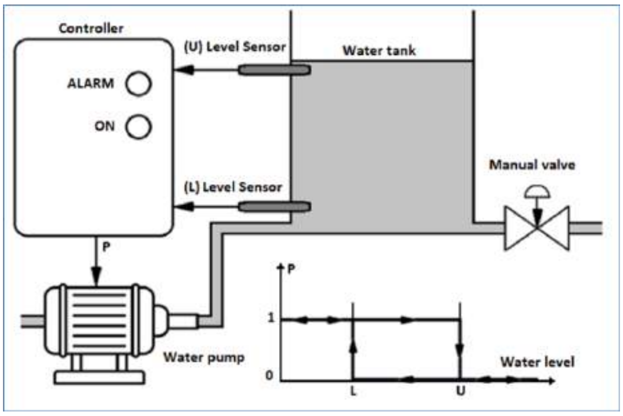
12. A serial communication channel that does not have a clock signal is said to be

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

13. The letter A in UART stands for

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

Q14—Q24: The figure to the right shows the connection between a water tank and its level Controller. The controller has two inputs: Upper-Level Sensor (U) and Lower-Level Sensor (L), which sense the level of the water inside the tank. Both sensors are connected through signal conditioning circuits to output logic high/low voltage levels. Controller output (P) is used to control a water pump. ON (Green) and ALARM (Red) LED indicators are used to show the state of the pump. The controller runs PROG1 (on the next page) on a TM4C microcontroller. The pump should be turned on (by applying a logic high on signal P), if the water level in the tank is below the low level until reaching the upper level, then it is turned off until the water level drops again below the lower level. If it happens that the pump is on for a specific time duration without having the water reaching the upper level, the pump is turned off and the ALARM LED is switched on for a specific duration. Then, the pump is switched on again (with ALARM LED off), continuing as normal.



14. In PROG1, line 1, constant CONST1 (used to reach the ALARM LED through PF1), should be

A) 0x01	B) 0x02	C) 0x03	D) 0x04

15. In PROG1, line 2, constant CONST2 (used to reach the water pump through PF2), should be

A) 0x01	B) 0x02	C) 0x03	D) 0x04

16. In PROG1, line 3, constant CONST3 (used to reach the ON LED through PF3), should be

A) 0x01	B) 0x04	C) 0x06	D) 0x08

17. In PROG1, line 5, condition C1 (used to switch pump states from PUMP_OFF to PUMP_ON) should be

A) !current_L && !old_L	B) !current_L && old_L	C) current_L && !old_L	D) current_L && old_L

18. In PROG1, line 8, condition C2 (used to switch pump states from PUMP_ON to PUMP_OFF) should be

A) !current_U && !old_U	B) !current_U && old_U	C) current_U && !old_U	D) current_U && old_U

19. In PROG1, line 9, condition C3 (used to switch pump states from PUMP_ON to PUMP_ALARM) should be

A) !current_L && on_state_counter>COUNTER_TH)	B) !current_U && on_state_counter>COUNTER_TH)
C) current_L && on_state_counter<COUNTER_TH)	D) current_U && on_state_counter<COUNTER_TH)

20. In PROG1, line 13, condition C4 (used to switch pump states from PUMP_ALARM to PUMP_ON) should be

A) current_U==1	B) old_U==0	C) on_state_counter<COUNTER_TH	D) on_state_counter>COUNTER_TH

21. In PROG1, line 6, V1, V2, V3 (used to set pins values) should be

A) OFF, OFF, OFF	B) OFF, OFF, ON	C) OFF, ON, OFF	D) ON, OFF, OFF

22. In PROG1, line 11, V4, V5, V6 (used to set pins values) should be

A) OFF, OFF, OFF	B) OFF, ON, ON	C) ON, ON, OFF	D) ON, OFF, ON

23. In PROG1, line 14, V7, V8, V9 (used to set pins values) should be

A) OFF, OFF, OFF	B) OFF, ON, ON	C) ON, ON, OFF	D) OFF, OFF, ON

24. In PROG1, line 19, statement <S1> (used to get pins values) should be

A) return ((GPIO_PORTF_DATA_R||PIN)<0);	B) return ((GPIO_PORTF_DATA_R&PIN)>0);
C) return ((GPIO_PORTF_DATA_R|PIN)==0);	D) return ((GPIO_PORTF_DATA_R&PIN)<0);

PROG1 Q14—Q24

```
1    #define L_PIN             0x01U          // PF0 is connected to Lower-Level (L) sensor
     #define ALARM_LED_PIN     <CONST1>       // PF1 is connected to ALARM LED (Red)
2    #define P_PIN             <CONST2>       // PF2 is connected to water pump
3    #define ON_LED_PIN        <CONST3>       // PF3 is connected to ON LED (Green)
4    #define U_PIN             0x10U          // PF4 is connected to Upper-Level (U) sensor
     #define COUNTER_TH        100000         // Threshold counter value
     #define PUMP_OFF          0
     #define PUMP_ON           1
     #define PUMP_ALARM        2
     #define ON                1
     #define OFF               0

     void Controller_Init(void);
     void Set_PIN(unsigned char, unsigned char);
     unsigned char Get_PIN(unsigned char);
     int main(void){
         unsigned char old_L, old_U; current_L, current_U, state;
         unsigned long on_state_counter;
         Controller_Init();
         state=PUMP_OFF;
         on_state_counter=0;
         for(;;){
             current_L=Get_PIN(L_PIN); current_U=Get_PIN(U_PIN);
             if (state==PUMP_OFF){
5                if (<C1>) state=PUMP_ON;
6                Set_PIN(P_PIN,V1); Set_PIN(ON_LED_PIN,V2); Set_PIN(ALARM_LED_PIN,V3);
7            } else if(state==PUMP_ON){
8                if (<C2>) state=PUMP_OFF;
9                if (<C3>){state=PUMP_ALARM; on_state_counter=0;}
10               if(!current_L) on_state_counter+=1;
11               Set_PIN(P_PIN,V4); Set_PIN(ON_LED_PIN,V5); Set_PIN(ALARM_LED_PIN,V6);
12           } else if(state==PUMP_ALARM){
                 on_state_counter+=1;
13               if (<C4>){state=PUMP_ON; on_state_counter=0;}
14               Set_PIN(P_PIN,V7); Set_PIN(ON_LED_PIN,V8); Set_PIN(ALARM_LED_PIN,V9);
15           }
             old_L=current_L; old_U=current_U;
         }
     }
16   void Controller_Init(void){
17       SYSCTL_RCGCGPIO_R |= 0x00000020;                    // Activate PORTF
         while((SYSCTL_PRGPIO_R&0x00000020)==0);
         GPIO_PORTF_DIR_R = 0x0E; GPIO_PORTF_PUR_R = 0x11; GPIO_PORTF_DEN_R = 0x1F;
     }
18   unsigned char Get_PIN(unsigned char PIN){
19       <S1>
     }
20   void Set_PIN(unsigned char PIN, unsigned char value){
21       if (value) GPIO_PORTF_DATA_R |= PIN;
         else GPIO_PORTF_DATA_R &= ~PIN;
     }
```

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

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

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

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

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

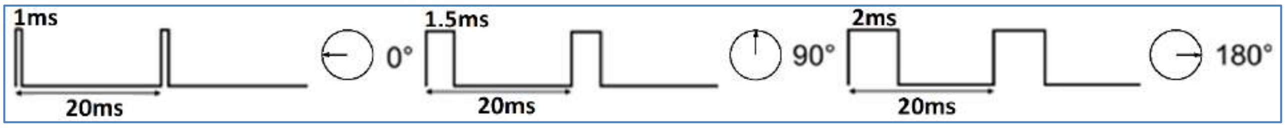
31. In PROG2, the UART is enabled in

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

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

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

Q33—Q38: The following figure illustrates the linear relationship between the input wave to a servo motor and the angle the motor rotates to. The servo motor controller runs PROG3 below on a TM4C microcontroller. Some statements in PROG3 are not in the right order.



PROG3 Q33—Q38

```
 1   #define MOTOR_PIN       0x02U
     #define ON              1
     #define OFF             0
     #define PERIOD_MS       20              // Period of the input wave to the servo motor
 2   void Controller_Init(void);
     void delay_ms(<T1> v);
     void Set_PIN(unsigned char, unsigned char);
     float angle_2_time(unsigned char);
     unsigned char Angles_Array []={00,45,90,180};

     int main(void){
         int i;
         <T1> delay_time; unsigned char angle;
         Controller_Init();
         for(;;){
             for (i=0;i<4;i++){
 3               delay_time=PERIOD_MS-delay_time; delay_ms(delay_time);
 4               delay_time=angle_2_time(angle); delay_ms(delay_time);
 5               Set_PIN(MOTOR_PIN,OFF);
 6               angle=Angles_Array[i];
 7               Set_PIN(MOTOR_PIN,ON);
 8           }
         }
     }
 9   <T1> angle_2_time(unsigned char angle){
10       return (<E1>);
11   }
12   void delay_ms(<T1> v){
13       unsigned long i=0;
14       unsigned long d=v*(<S1>);
15       while(i<d) i++;
     }
16   void Set_PIN(unsigned char PIN, unsigned char val){
         if (val) GPIO_PORTF_DATA_R |= PIN;
         else GPIO_PORTF_DATA_R &= ~PIN;
     }
     void Controller_Init(void){
         SYSCTL_RCGCGPIO_R |= 0x00000020;            // Activate PORTF
         while((SYSCTL_PRGPIO_R&0x00000020)==0);
         GPIO_PORTF_DIR_R=0x0E;                      // PF0 and PF4 as input and PF3-1 as output
         GPIO_PORTF_PUR_R=0x11;                      // Enable pull-up on PF0 and PF4
         GPIO_PORTF_DEN_R=0x1F;                      // Enable digital I/O on PF4-0
     }
```

33. In PROG3, line 10, equation <E1> (used to convert an angle to input pulse duration in ms) should be

A) 1-angle*(1.0/180)	B) 1+angle*(2.0/360)	C) 1+angle*(1/180)	D) 1+angle*(2.0/180)

34. In PROG3, if the input pulse duration is 1.75 ms, the angle the motor will rotate to will be

A) -135°	B) -67.5°	C) 67.5°	D) 135°

35. In PROG3, block lines 2 and lines 9 and 12, datatype <T1> should best be set to

A) int	B) unsigned int	C) long	D) float

36. In PROG3, if the controller uses a 16-MHz clock, function delay_ms(1.0) is called to insert a 1-ms delay, and a single loop iteration at line 15 takes 5 cycles, integer value <S1> in line 14 should be

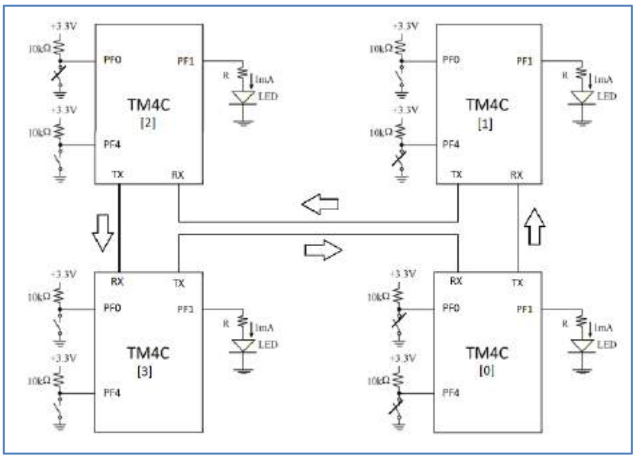
A) 80	B) 800	C) 1600	D) 3200

37. In PROG3, assume that <S1> is set as in the previous question and that the loop at line 15 is changed so that a single iteration takes 6 cycles instead of 5. If function delay_ms(1.5) is called to produce a 1.5-ms pulse to have the motor rotate 90°, the actual angle the motor will rotate to will be

A) 36°	B) 91°	C) 95°	D) 144°

38. In PROG3, the correct order for lines 3—7 should be

A) 6-7-3-5-4	B) 6-7-4-5-3	C) 7-6-4-5-3	D) 6-5-4-7-3



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

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

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

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>

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>

43. In PROG4, the functionality of line 14 is to wait until

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;
     }
```

على طلبة قسم قوى الإجابة على الأسئلة 45—60 فى الصفحات 8—9 فقط.

على طلبة قسم حاسبات وطلبة قسم إتصالات الإجابة على الأسئلة 45—60 فى الصفحات 10—12 فقط.

PROG5: Q45—Q48 **سؤال مخصص لطلبة قسم قوى فقط.**

```
1  #include <stdio.h>
2  int main(){
3      int i, j, temp;
4      int a[4] = {10,2,0,14};
5      for(i = 0; i<4; i++){
6          for(j = i; j<4; j++){
7              if(a[j] = a[i]){
8                  a[i] = a[j]; temp = a[i]; a[j] = temp;
9              }
10         }
11     }
12     for(i = 0; i<4; i++) printf("%d-",a[i]);
13 }
```

45. In PROG5, to sort the array faster, line 6 should be:

A) for(j = i+1; j<4; j++){	B) for(j = i-1; j<4; j++){	C) for(j = 1; j<4; j++){	D) for(j = 0; j<4; j++){

46. In PROG5, line 7, condition is not correct and it should be:

A) a[j]!=a[i]	B) a[j]>a[i]	C) a[j]==a[i]	D) a[j]<a[i]

47. In PROG5, line 8 is not correct and it should be:

A) a[j] = a[i]; a[j] = temp; temp = a[i];	B) temp = a[i]; a[i] = a[j]; a[j] = temp;
C) temp = a[i]; a[j] = a[i]; a[j] = temp;	D) a[j] = temp; temp = a[i]; a[i] = a[j];

48. In PROG5, after fixing the program, how many times will line 7 be executed?

A) 4	B) 6	C) 7	D) 8

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 قوى فقط.**

```
1  #define GPIO_PF0_M        0x01U                 // PF0 is connected to direction switch
   #define STEPPER           GPIO_PORTD_DATA_R
2  void Controller_Init(void);                     // Initialize PORTF and PORTD
   unsigned char SW_Input(void);                   // Read the state of PF0 (switch)
   void delay1ms(long delay);                      // 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°

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

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

54. In PROG6, line 8, statement <S4> should be

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

Q56—Q60: PROG7 changes a text from lower case to upper case. Some statements in it are not correct.

PROG7 Q56—Q60 سؤال مخصص لطلبة قسم قوى فقط.

```
1  #include <stdio.h>
2  void cap(char *p);
3  int main(){
4      char text[8]="Ain-Shams";
5      cap(text); printf("%s",text);
6  }
7  void cap(char *p){
8      char v=0x41;
9      while(p==0){
10         if ((*p<'A') && (*p>'Z'))
11             *p=(v);
12         p++;
13     }
   }
```

56. In PROG7, line 4 is not correct and it should be:

A) char text*[9]="Ain-Shams";	B) char text&[8]="Ain-Shams";
C) char text[10]="Ain-Shams";	D) char text&[9]="Ain-Shams";

57. In PROG7, given that ASCII of letter 'A' is 65 and ASCII of 'a' is 97, line 8 is not correct and it should be:

A) char v=0x01;	B) char v=0x10;	C) char v=0x20;	D) char v=0x30;

58. In PROG7, line 9, condition is not correct and it should be:

A) *p==0	B) *p=0	C) *p!=0	D) p!=0

59. In PROG7, line 10, condition is not correct and it should be:

A) (*p<='A') && (*p<='Z')	B) (*p>='a') && (*p<='z')	C) (*p<='A') && (*p>='Z')	D) (*p<='a') && (*p<='z')

60. In PROG7, after fixing the program as in previous questions, line 11 is not correct and it should be:

A) *p |=(v);	B) *p |=~(v);	C) *p &=(v);	D) *p &=~(v);

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.

```
    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){
1     Controller_Init(); UART_Init();
      UART_OutChar('A');
      for(;;);
      UART_OutChar('C');
    }
    void GPIOF_Handler(void){
      UART_OutChar('B');
2     GPIO_PORTF_ICR_R=0x10;
    }
    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
3     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

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
3           add     sp, #s_size
4           ldr     r0, =a
5           mov     r4, #3
6           str     r4, [r0]
7           ldr     r1, =b
8           mov     r4, #5
9           str     r4, [r1]
10          ldr     r2, [r0]
11          ldr     r3, [r1]
12          add     r2, #1
13          add     r3, #1
14          bl      func
15          b       stop
16  func
            push    {r2-r3}
17          ldr     r2, [r0]
18          ldr     r3, [r1]
19          str     r2, [r1]
20          str     r3, [r0]
21          pop     {r2-r3}
22          bx      lr
    stop
            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	 A Z 	AREA READ_variables, DATA, READONLY DCD 5 AREA WRITE_variables, DATA, READWRITE DCD 0 AREA MYCODE, CODE, READONLY
2		LDR r0, =A
3		LDR r1, [r0]
4		MOV r2, #1
5		CMP r1, #0
6		BLE LOC2
7	LOC1	 MUL r3, r2, r1
8		MOV r2, r3
9		SUB r1, r1, #1
10		CMP r1, #0
11		BLE LOC2
12		B LOC1
13	LOC2	 LDR r4, =Z
14		STR r2, [r4] 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