AIN SHAMS UNIVERSITY FACULTY OF ENGINEERING CREDIT HOURS ENGINEERING PROGRAMS		
June 2nd,2018	**Course Code: CSE 345&347**	***Time: 3 Hours***
Real-Time and Embedded Systems Design		
The Exam Consists of **6** Questions in **6** Pages	***Total Marks: 40 Marks***	**4/6**

Question (4):(8 Marks)

The figure below is a snap shot from a debugging session. In the following table, document your expectation of the hitting-order of Breakpoints (designated by line number). Consider multiple hits of a break point, inside a loop, as ONLY 1 HIT.

```
63  int main( void )
64  {
65    xMutex = xSemaphoreCreateMutex();
66    if( xMutex != NULL )
67    {
68      xTaskCreate( prvPrintTask, "Print1", 240, "Task 1 **\n", 1, NULL );
69      xTaskCreate( prvPrintTask, "Print2", 240, "Task 2 --\n", 2, NULL );
70      vTaskStartScheduler();
71    }
72  }
73  static void prvNewPrintString( const portCHAR *pcString )
74  {
75  static char cBuffer[ mainMAX_MSG_LEN ];
76  int i,j;
77    xSemaphoreTake( xMutex, portMAX_DELAY );
78    {
79      sprintf( cBuffer, "%s", pcString );
80      for (i=1;i<1000000;i++)
81      {
82        j++;
83      }
84      printf( cBuffer );
85    }
86    xSemaphoreGive( xMutex );
87  }
88  static void prvPrintTask( void *pvParameters )
89  {
90  char *pcStringToPrint;
91  int i,j;
92    pcStringToPrint = ( char * ) pvParameters;
93    for( ;; )
94    {
95      prvNewPrintString( pcStringToPrint );
96      for (i=1;i<1000000;i++)
97      {
98        j++;
99      }
100     vTaskDelay( 255 );
101   }
102 }
```

Breakpoint 77 · Breakpoint 79 · Breakpoint 82 · Breakpoint 84 · Breakpoint 86 · Breakpoint 95 · Breakpoint 98 · Breakpoint 100

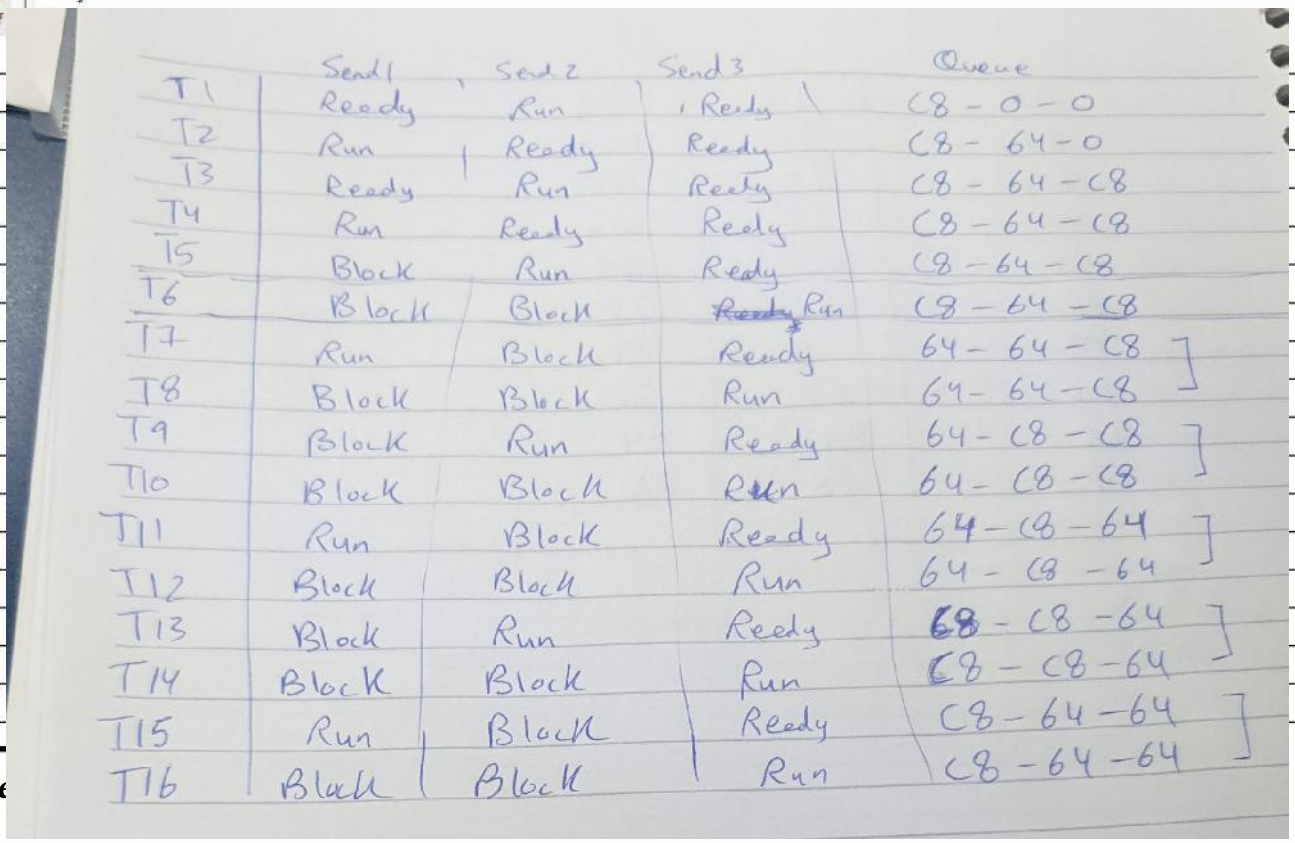
Hit Order	Break Point No.	Hit Order	Break Point No.	Hit Order	Break Point No.
1	95	***11***	79	***21***	98
2	77	***12***	82	***22***	100
3	79	***13***	95	***23***	98
4	82	***14***	77	***24***	95
5	84	***15***	84	***25***	77
6	86	***16***	86	***26***	79
7	98	***17***	79	***27***	82
8	100	***18***	82	***28***	84
9	95	***19***	84	***29***	86
10	77	***20***	86	***30***	98

AIN SHAMS UNIVERSITY FACULTY OF ENGINEERING CREDIT HOURS ENGINEERING PROGRAMS		
June 2nd,2018	**Course Code: CSE 345&347**	***Time: 3 Hours***
Real-Time and Embedded Systems Design		
The Exam Consists of **6** Questions in **6** Pages	***Total Marks: 40 Marks***	**5/6**

Question (5): (8 Marks)

The figure below is a snap shot from FreeRTOS application using tasks communicating through a queue. Fill the given table with task states (Ready, Running, or Blocked) in consecutive time slots Tx. Show the content of the queue at the end of each slot assuming initial HEX value of (00 00 00 00 – 00 00 00 00 – 00 00 00 00). "Sender 2" is the first running task just after scheduler-start in T1.

```
60  int main( void )
61  {
62      xQueue = xQueueCreate( 3, sizeof( long ) );
63      xTaskCreate( vSenderTask, "Sender1", 240, ( void * ) 100, 2, NULL );
64      xTaskCreate( vSenderTask, "Sender2", 240, ( void * ) 200, 2, NULL );
65      xTaskCreate( vReceiverTask, "Receiver", 240, NULL, 1, NULL );
66      vTaskStartScheduler();
67  }
68  static void vSenderTask( void *pvParameters )
69  {
70  long lValueToSend;
71  const portTickType xTicksToWait = 100 / portTICK_RATE_MS;
72  lValueToSend = ( long ) pvParameters;
73    for( ;; )
74    {
75      xQueueSendToBack( xQueue, &lValueToSend, xTicksToWait );
76      taskYIELD();
77    }
78  }
79  static void vReceiverTask( void *pvParameters )
80  {
81  long lReceivedValue;
82    for( ;; )
83    {
84      xQueueReceive( xQueue, &lReceivedValue, 0 );
85      vPrintStringAndNumber( "Received = ", lReceivedValue );
86    }
87
```



	Send1	Send2	Send3	Queue
T1	Ready	Run	Ready	C8 - 0 - 0
T2	Run	Ready	Ready	C8 - 64 - 0
T3	Ready	Run	Ready	C8 - 64 - C8
T4	Run	Ready	Ready	C8 - 64 - C8
T5	Block	Run	Ready	C8 - 64 - C8
T6	Block	Block	~~Ready~~ Run	C8 - 64 - C8
T7	Run	Block	Ready	64 - 64 - C8
T8	Block	Block	Run	64 - 64 - C8
T9	Block	Run	Ready	64 - C8 - C8
T10	Block	Block	Run	64 - C8 - C8
T11	Run	Block	Ready	64 - C8 - 64
T12	Block	Block	Run	64 - C8 - 64
T13	Block	Run	Ready	68 - C8 - 64
T14	Block	Block	Run	C8 - C8 - 64
T15	Run	Block	Ready	C8 - 64 - 64
T16	Block	Block	Run	C8 - 64 - 64

AIN SHAMS UNIVERSITY FACULTY OF ENGINEERING CREDIT HOURS ENGINEERING PROGRAMS		
June 2nd,2018	**Course Code: CSE 345&347**	***Time: 3 Hours***
Real-Time and Embedded Systems Design		
The Exam Consists of **6** Questions in **6** Pages	***Total Marks: 40 Marks***	**6/6**

Question (6):(8 Marks)

The figure below is a snap shot from a debugging session. Show the sequence of execution timing diagram starting just after vTaskStartScheduler. Tasks should be shown down-up vertically according to their priorities (e.g. Highest priority should be on the top). Also show the debug (printf) viewer.

```
82  int main( void )
83  {
84      vSemaphoreCreateBinary( xBinarySemaphore );
85      if( xBinarySemaphore != NULL )
86      {
87          prvSetupSoftwareInterrupt();
88          xTaskCreate( vHandlerTask, "Handler", 240, NULL, 1, NULL );
89          xTaskCreate( vPeriodicTask, "Periodic", 240, NULL, 3, NULL );
90          vTaskStartScheduler();
91      }
92  }
93  static void vHandlerTask( void *pvParameters )
94  {
95      xSemaphoreTake( xBinarySemaphore, 0 );
96      for( ;; )
97      {
98          xSemaphoreTake( xBinarySemaphore, portMAX_DELAY );
99          vPrintString( "Handler task - Processing event.\n" );
100     }
101 }
102 static void vPeriodicTask( void *pvParameters )
103 {
104     for( ;; )
105     {
106         vTaskDelay( 500 / portTICK_RATE_MS );
107         vPrintString( "Periodic task - About to generate an interrupt.\n" );
108         mainTRIGGER_INTERRUPT();
109         vPrintString( "Periodic task - Interrupt generated.\n\n" );
110     }
111 }
112 void vSoftwareInterruptHandler( void )
113 {
114 portBASE_TYPE xHigherPriorityTaskWoken = pdFALSE;
115     xSemaphoreGiveFromISR( xBinarySemaphore, &xHigherPriorityTaskWoken );
116     mainCLEAR_INTERRUPT();
117     portEND_SWITCHING_ISR( xHigherPriorityTaskWoken );
118 }
```