AIN SHAMS UNIVERSITY
FACULTY OF ENGINEERING
ICHEP; MCT-CESS
Spring 2022

June 9th,2022	Course Code: CSE411-345-347 Real-Time & Embedded Systems Design	Time: 2 Hours

The Exam Consists of **5 Questions in 5 Pages** **Total Marks: 40 Marks**

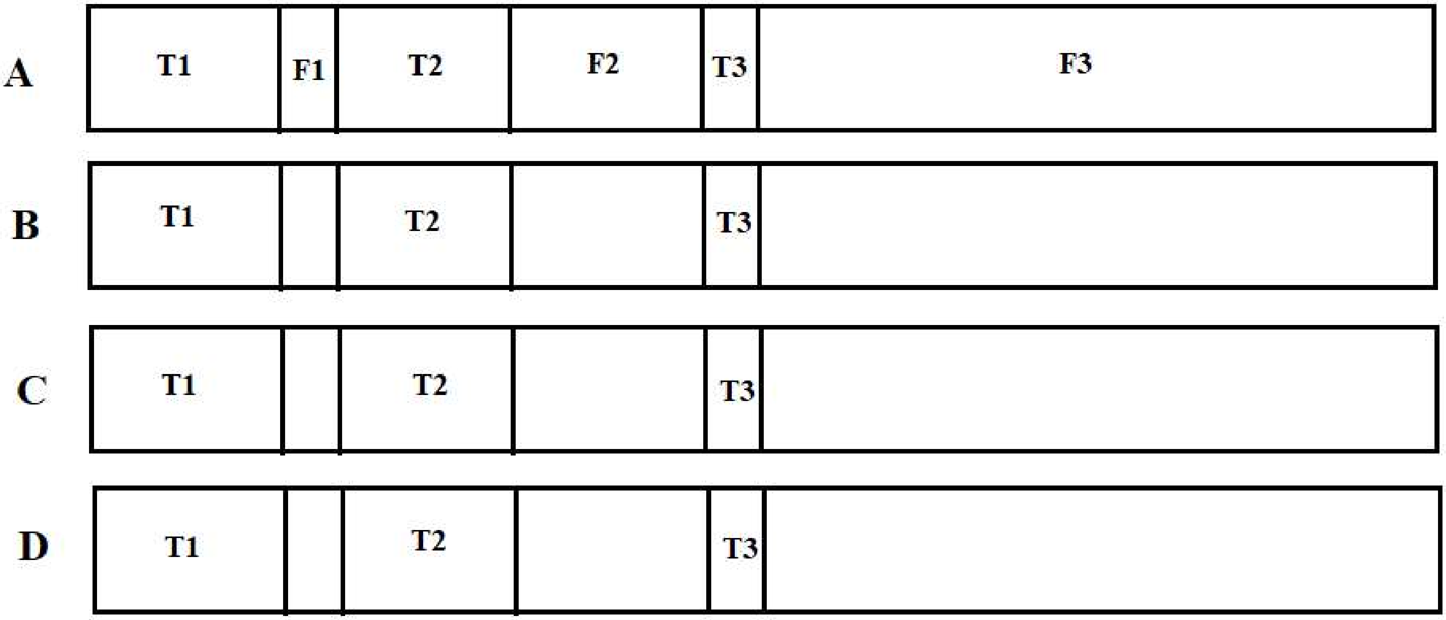
تنبيه هام جدا: يجب على كل طالب **الحل فقط هنا في ورق الاسئلة** ولن يلتفت لأي إجابة في الكراسة المرفقة والتي تحتوي على البار كود الخاص بكل طالب

Important Rules:	تعليمات هامة
- Having a (mobile -Smart Watch- earphones) inside the examination hall is forbidden and is considered as a cheating behavior. - It is forbidden to have any references, notes, books, or any other materials even if it is not related to the exam content with you in the examination hall. - This is an answer sheet. - Assume missing data if any – Read it all well, at first.	- حيازة (المحمول- الساعات الذكية - سماعة الأذن) داخل لجنة الامتحان يعتبر حالة غش تستوجب العقاب . - لايسمح بدخول أي كتب أو ملازم أو أوراق داخل اللجنة والمخالفة تعتبر حالة غش. - هذه ورقة إجابة أيضا - أقرأها أولا جيدا – أفترض الناقص إن وجد

Question 1: (3 Marks)

For FreeRTOS based embedded hypothetical application, “heap2” Heap memory algorithm is used. Figure A shows heap memory state where T1, T2 and T3 are allocated tasks. F1, F2, and F3 are free spaces of 100 bytes, 400 bytes, and 900 bytes respectively. **Show in the following figures, heap memory states for the cases:**

B: Create T4 of 600 bytes (Stack and TCB)
C: Create T5 of 350 bytes (Stack and TCB)
D: Create T6 of 36 bytes (Stack and TCB)

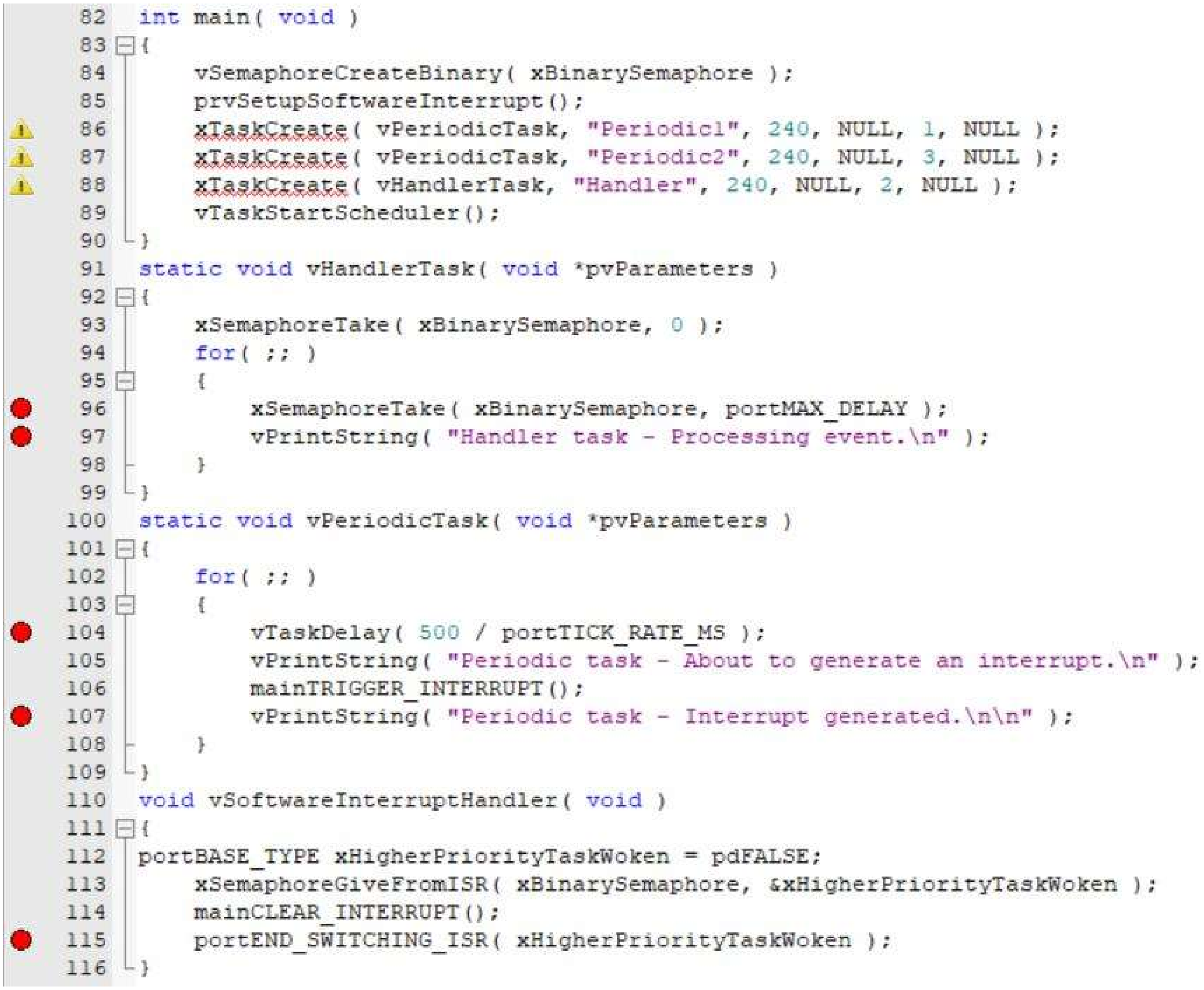


Question 2: (10 Marks) Example 12

Assume the following snippet of code/application that already had all necessary declarations, inclusions, and prototypes. **In the given table, order the first 10 break points to be hit, when GO is pressed.**

1st Break Point Hit	2nd Break Point Hit	3rd Break Point Hit	4th Break Point Hit	5th Break Point Hit
104	96	104	115	107

6th Break Point Hit	7th Break Point Hit	8th Break Point Hit	9th Break Point Hit	10th Break Point Hit
104	97	96	115	97



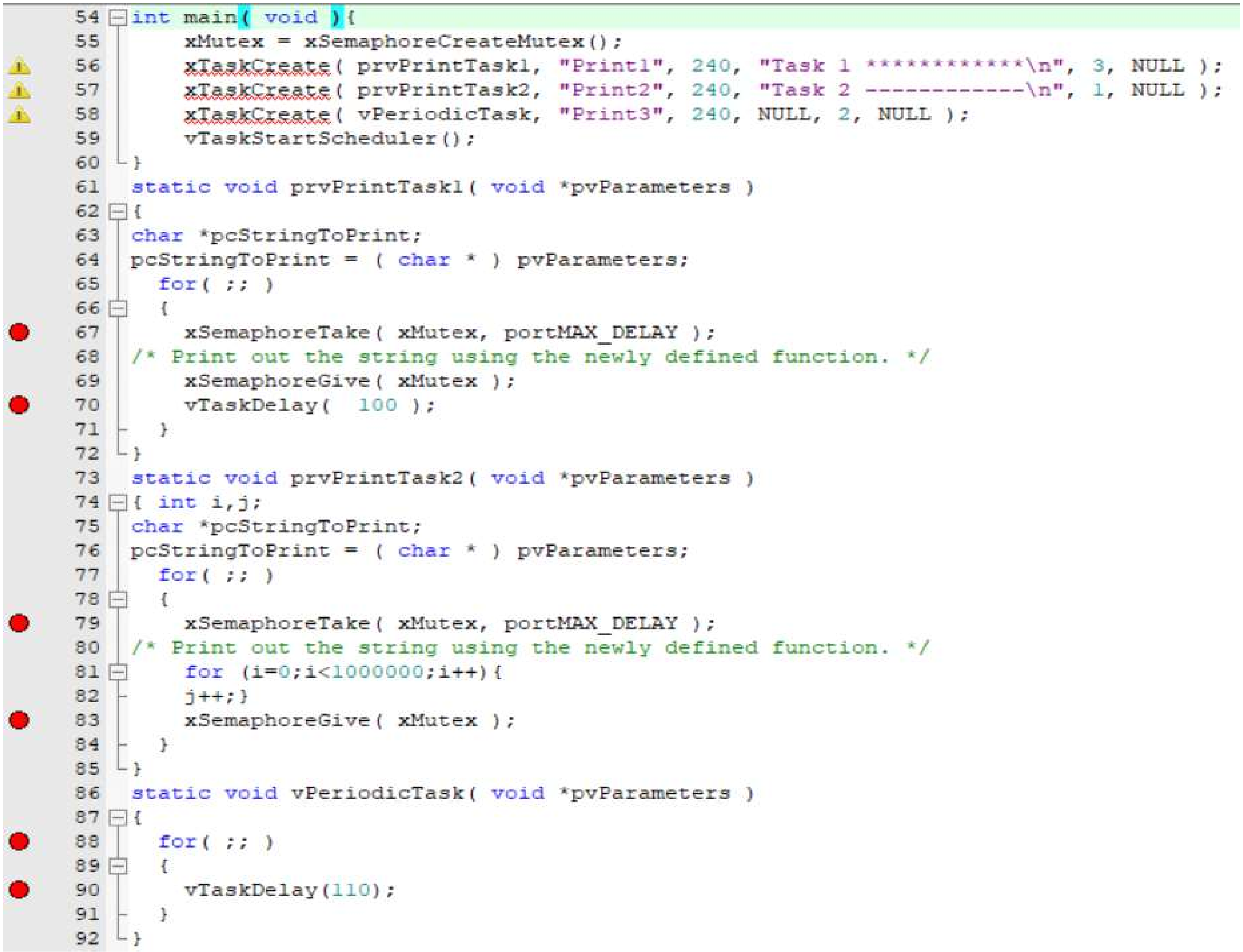
```
     82  int main( void )
     83  {
     84      vSemaphoreCreateBinary( xBinarySemaphore );
     85      prvSetupSoftwareInterrupt();
⚠    86      xTaskCreate( vPeriodicTask, "Periodic1", 240, NULL, 1, NULL );
⚠    87      xTaskCreate( vPeriodicTask, "Periodic2", 240, NULL, 3, NULL );
⚠    88      xTaskCreate( vHandlerTask, "Handler", 240, NULL, 2, NULL );
     89      vTaskStartScheduler();
     90  }
     91  static void vHandlerTask( void *pvParameters )
     92  {
     93      xSemaphoreTake( xBinarySemaphore, 0 );
     94      for( ;; )
     95      {
●    96          xSemaphoreTake( xBinarySemaphore, portMAX_DELAY );
●    97          vPrintString( "Handler task - Processing event.\n" );
     98      }
     99  }
    100  static void vPeriodicTask( void *pvParameters )
    101  {
    102      for( ;; )
    103      {
●   104          vTaskDelay( 500 / portTICK_RATE_MS );
    105          vPrintString( "Periodic task - About to generate an interrupt.\n" );
    106          mainTRIGGER_INTERRUPT();
●   107          vPrintString( "Periodic task - Interrupt generated.\n\n" );
    108      }
    109  }
    110  void vSoftwareInterruptHandler( void )
    111  {
    112  portBASE_TYPE xHigherPriorityTaskWoken = pdFALSE;
    113      xSemaphoreGiveFromISR( xBinarySemaphore, &xHigherPriorityTaskWoken );
    114      mainCLEAR_INTERRUPT();
●   115      portEND_SWITCHING_ISR( xHigherPriorityTaskWoken );
    116  }
```

Question 3: (10 Marks) Example 14 & 15

Assume the following snippet of code/application that already had all necessary declarations, inclusions, and prototypes. **In the given table, order the first 10 break points to be hit, when GO is pressed.**

1st Break Point Hit	2nd Break Point Hit	3rd Break Point Hit	4th Break Point Hit	5th Break Point Hit
67	70	88	90	79

6th Break Point Hit	7th Break Point Hit	8th Break Point Hit	9th Break Point Hit	10th Break Point Hit
67	83	70	90	79



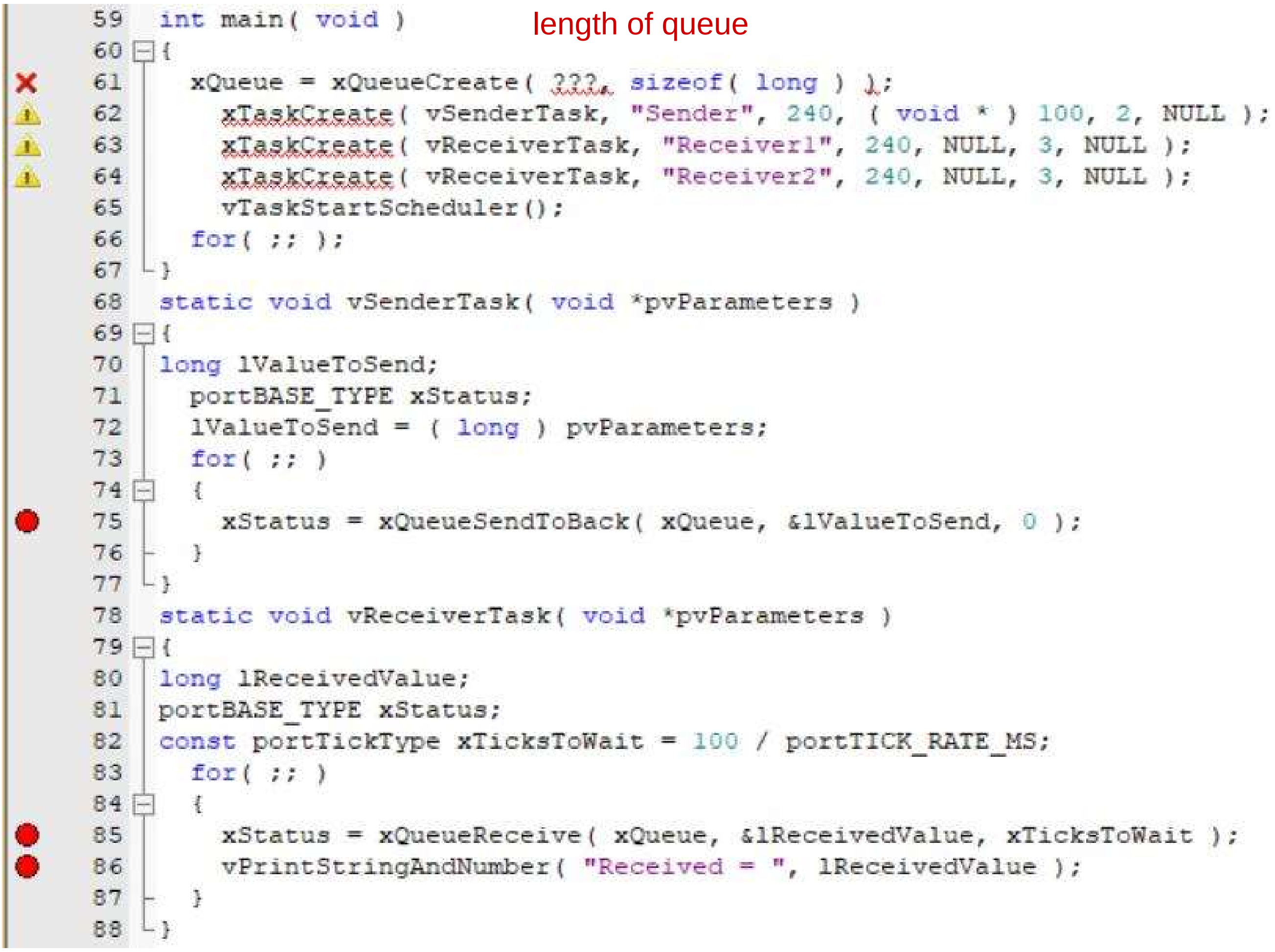
```
   54 int main( void ){
   55     xMutex = xSemaphoreCreateMutex();
⚠  56     xTaskCreate( prvPrintTask1, "Print1", 240, "Task 1 ************\n", 3, NULL );
⚠  57     xTaskCreate( prvPrintTask2, "Print2", 240, "Task 2 ------------\n", 1, NULL );
⚠  58     xTaskCreate( vPeriodicTask, "Print3", 240, NULL, 2, NULL );
   59     vTaskStartScheduler();
   60 }
   61 static void prvPrintTask1( void *pvParameters )
   62 {
   63 char *pcStringToPrint;
   64 pcStringToPrint = ( char * ) pvParameters;
   65   for( ;; )
   66   {
●  67     xSemaphoreTake( xMutex, portMAX_DELAY );
   68 /* Print out the string using the newly defined function. */
   69     xSemaphoreGive( xMutex );
●  70     vTaskDelay(  100 );
   71   }
   72 }
   73 static void prvPrintTask2( void *pvParameters )
   74 { int i,j;
   75 char *pcStringToPrint;
   76 pcStringToPrint = ( char * ) pvParameters;
   77   for( ;; )
   78   {
●  79     xSemaphoreTake( xMutex, portMAX_DELAY );
   80 /* Print out the string using the newly defined function. */
   81     for (i=0;i<1000000;i++){
   82     j++;}
●  83     xSemaphoreGive( xMutex );
   84   }
   85 }
   86 static void vPeriodicTask( void *pvParameters )
   87 {
●  88   for( ;; )
   89   {
●  90     vTaskDelay(110);
   91   }
   92 }
```

Question 4: (9 Marks) Example 10

Assume the following snippet of code/application that already had all necessary declarations, inclusions, and prototypes. **In the given table, order the first 8 break points (PB) to be hit, when GO is pressed. At each PB, Define the states (READY, RUNNING or BLOCK) of all tasks. (Hint: "Receiver 2" will be the first to execute once Scheduler starts).**

WHAT IS THE BEST YOU CAN CODE IN LINE 61, instead of "???", TO COMPILE? length of queue will be 1

Break Point at Line?	State of Sender	State of Receiver 1	State of Receiver 2
85	Ready	Ready	Running
85	Ready	Running	Blocked
75	Running	Blocked	Blocked
86	Ready	Blocked	Running
85	Ready	Blocked	Running
75	Running	Blocked	Blocked
86	Ready	Running	Blocked
85	Ready	Running	Blocked

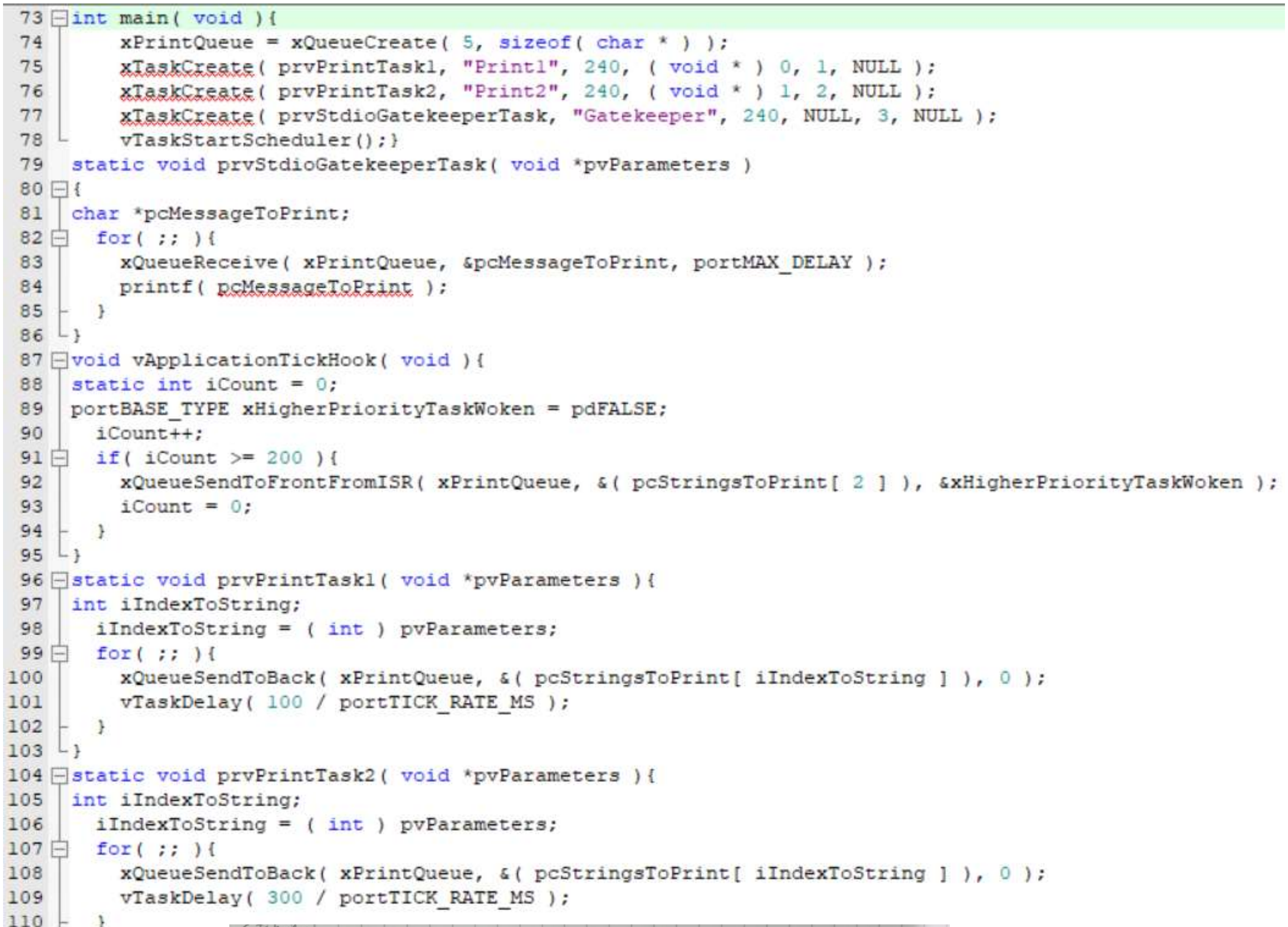


length of queue

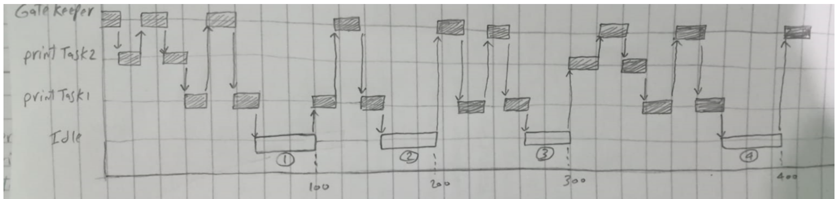
```
59  int main( void )
60  {
61    xQueue = xQueueCreate( ???, sizeof( long ) );
62      xTaskCreate( vSenderTask, "Sender", 240, ( void * ) 100, 2, NULL );
63      xTaskCreate( vReceiverTask, "Receiver1", 240, NULL, 3, NULL );
64      xTaskCreate( vReceiverTask, "Receiver2", 240, NULL, 3, NULL );
65      vTaskStartScheduler();
66    for( ;; );
67  }
68  static void vSenderTask( void *pvParameters )
69  {
70  long lValueToSend;
71    portBASE_TYPE xStatus;
72    lValueToSend = ( long ) pvParameters;
73    for( ;; )
74    {
75      xStatus = xQueueSendToBack( xQueue, &lValueToSend, 0 );
76    }
77  }
78  static void vReceiverTask( void *pvParameters )
79  {
80  long lReceivedValue;
81  portBASE_TYPE xStatus;
82  const portTickType xTicksToWait = 100 / portTICK_RATE_MS;
83    for( ;; )
84    {
85      xStatus = xQueueReceive( xQueue, &lReceivedValue, xTicksToWait );
86      vPrintStringAndNumber( "Received = ", lReceivedValue );
87    }
88  }
```

Question 5: (8 Marks)

For the following FreeRTOS based application snippet, **Sketch tasks timing diagram starting from 0ms and ending at 400ms.** Vertical axes should be graded down-up from low- to high-priority.



```
int main( void ){
    xPrintQueue = xQueueCreate( 5, sizeof( char * ) );
    xTaskCreate( prvPrintTask1, "Print1", 240, ( void * ) 0, 1, NULL );
    xTaskCreate( prvPrintTask2, "Print2", 240, ( void * ) 1, 2, NULL );
    xTaskCreate( prvStdioGatekeeperTask, "Gatekeeper", 240, NULL, 3, NULL );
    vTaskStartScheduler();}
static void prvStdioGatekeeperTask( void *pvParameters )
{
char *pcMessageToPrint;
  for( ;; ){
    xQueueReceive( xPrintQueue, &pcMessageToPrint, portMAX_DELAY );
    printf( pcMessageToPrint );
  }
}
void vApplicationTickHook( void ){
static int iCount = 0;
portBASE_TYPE xHigherPriorityTaskWoken = pdFALSE;
  iCount++;
  if( iCount >= 200 ){
    xQueueSendToFrontFromISR( xPrintQueue, &( pcStringsToPrint[ 2 ] ), &xHigherPriorityTaskWoken );
    iCount = 0;
  }
}
static void prvPrintTask1( void *pvParameters ){
int iIndexToString;
  iIndexToString = ( int ) pvParameters;
  for( ;; ){
    xQueueSendToBack( xPrintQueue, &( pcStringsToPrint[ iIndexToString ] ), 0 );
    vTaskDelay( 100 / portTICK_RATE_MS );
  }
}
static void prvPrintTask2( void *pvParameters ){
int iIndexToString;
  iIndexToString = ( int ) pvParameters;
  for( ;; ){
    xQueueSendToBack( xPrintQueue, &( pcStringsToPrint[ iIndexToString ] ), 0 );
    vTaskDelay( 300 / portTICK_RATE_MS );
  }
}
```



① Idle Task

During it → App Tick Hook will happen 100 Times

Count at the end = 100

② Idle Task

- During it → App Tick Hook will happen 100 Times

Count = 200 then = 0

③ Idle Task

During it → App Tick Hook will happen 100 Times

Count at the end = 100

④ Idle Task

During it → App ———— "

Count = 200 then 0