1. (10 pts) In the box below, write the output generated by this YAK application code when it executes. Assume that all semaphore references are shown and that all relevant variables and macros are defined correctly elsewhere in the code.

```
#define FLAG1 0x01
#define FLAG2 0x02

void main(void)
{
   YKInitialize();
   YKNewTask(Task1,(void *)&Stk1[SSIZE],10);
   YKNewTask(Task2,(void *)&Stk2[SSIZE],15);
   YKNewTask(Task3,(void *)&Stk3[SSIZE],20);
   SemA = YKSemCreate(0);
   Ev1 = YKEventCreate(0);
   YKRun();
}

void Task1(void)
{
   printString("A");
   YKEventPend(Ev1,FLAG1,EVENT_WAIT_ANY);
   YKEventReset(Ev1,FLAG1);
   printString("B");
   YKSemPend(SemA);
   printString("C");
   while (1)
   {
      YKDelayTask(5);
      /* misc. code, no output */
   }
}

void Task2(void)
{
   printString("D");
   YKNewTask(Task4,(void *)&Stk4[SSIZE],25);
   printString("E");
   YKSemPend(SemA);
   printString("F");
   while (1)
   {
      YKDelayTask(5);
      /* misc. code, no output */
   }
}
```

```
void Task3(void)
{
   printString("G");
   YKEventPend(Ev1,FLAG1|FLAG2,EVENT_WAIT_ALL);
   YKEventReset(Ev1,FLAG2);
   printString("H");
   YKSemPost(SemA);
   printString("I");
   while (1)
   {
      YKDelayTask(5);
      /* misc. code, no output */
   }
}

void Task4(void)
{
   printString("J");
   YKEventSet(Ev1,FLAG1|FLAG2);
   printString("K");
   YKSemPost(SemA);
   printString("L");
   while (1)
   {
      YKDelayTask(5);
      /* misc. code, no output */
   }
}
```

PROGRAM OUTPUT:

2. (5 pts) What are *timer callback functions*, why are they useful, and how might they be implemented?

3. (10 pts) Circle each function below whose normal implementation includes a call to the scheduler.

YKDelayTask YKQCreate YKNewTask YKEventSet YKEventPend YKQPend YKSemPost YKInitialize YKSemCreate YKEventReset

4. (10 pts) Circle each function below whose execution can cause a task to be added to the ready list.

YKDelayTask YKQCreate YKNewTask YKEventSet YKEventPend YKQPend YKSemPost YKInitialize YKSemCreate YKEventReset

5. (4 pts) What are the two rules that ISRs in an RTOS environment must follow that do not apply to task code?

6. (40 pts) Mark each of the following true or false by circling **T** or **F** respectively.

a. **T F** Some of the best candidates for encapsulation are accesses to shared data.

b. **T F** Studies show as much as 3x difference in productivity based on how often programmers are interrupted.

c. **T F** The RTOS is aware of which shared resource a particular semaphore protects.

d. **T F** Clock jitter arises because heartbeat timer interrupts are not asserted at precise intervals.

e. **T F** It is common for embedded microprocessors to contain timers unrelated to the heartbeat timer.

f. **T F** In YAK, kernel code never dereferences the void pointers in a message queue.

g. **T F** In a rate monotonic system, the task assigned the highest priority is the task

expected to run most frequently.

h. **T F** In any real-time system with rate monotonic priority assignment, all deadlines are guaranteed to be met.

i. **T F** A nonconforming interrupt routine does not save or restore register context when it runs.

j. **T F** In most embedded systems, a loader places the program in memory and performs all required address fix ups.

k. **T F** In all common power-saving modes, suspended processors require a reset to start up again.

l. **T F** A monitor is a program that resides on the target that can receive and run new programs on the target.

m. **T F** To use assert macros on your target, you may have to write a custom routine to stop normal execution.

n. **T F** Increasing the number of tasks is likely to increase the need for semaphores and message queues.

o. **T F** Instruction set simulators can help determine response time and throughput in application code.

p. **T F** Lockheed-Martin's space shuttle code averaged less than l error per 100,000 lines of source code.

q. **T F** The lack of mutual exclusion on shared data accesses was an important factor in the Therac-25 accidents.

r. **T F** Some trading firms have server farms near stock exchanges to reduce the delay for their electronic orders.

s. **T F** Prof. Edward Lee argued that core abstractions in computing must be changed to reflect the passage of time.

t. **T F** The paper on memory tests suggested separate tests for data, address, and control lines.

7. (5 pts) What particular challenge does *initialized data* present in embedded systems, how is it dealt with, and why does the challenge not arise in conventional (desktop) systems?

8. (6 pts) Assuming a YAK-like RTOS, what are the major differences between using *semaphores* and using *events*?

9. (10 pts) Identify and briefly describe five specific memory segments (by category) that necessarily are handled differently by the linker/locator.