1. Drag and drop the words in the correct blanks. The question is about Remote Procedure Calls (RPCs).
   When the user calls the kernel to send RPC message to a procedure, the kernel then sends a message to **matchmaker** to find **port number**. An answer is now received by the kernel (client), which then places the answer in **user RPC message** and the kernel actually sends a **remote procedure call**. This is then received by a **daemon** that is listening and it processes the request and sends the output back to the kernel, which then passes the reply to the user.

2. Match the following regarding pthread condition variables: Function mapped to the description.
   pthread_cond_init: **create a condition variable**
   pthread_cond_destroy: **kill a condition variable**
   pthread_cond_wait: **block waiting for a signal**
   pthread_cond_signal: **signal another thread and wake it up**
   pthread_cond_broadcast: **signal multiple threads and wake them all up**

3. Choose all the options that are true about semaphores.
   Select one or more:
   a. **A basic semaphore can be implemented using an integer variable.**
   b. The value of a binary semaphore can take any real value between 0 and 1.
   c. **Counting semaphores can range over an unrestricted domain.**
   d. **Two processes cannot execute wait() and signal() operations on the same semaphore at the same time.**
   e. SInce semaphores are managed by the OS, two processes can issue wait() or signal() operations on the same semaphore at the same time.
   f. If a semaphore is implemented using a waiting queue, deadlocks can occur between processes because of the signal() event.

4. Select the best answer with respect to deadlock elimination.
   a. We can eliminate deadlock by terminating all the processes that are deadlocked.
   b. We can eliminate deadlock by terminating one process at a time until a deadlock cycle is eliminated.
   c. We can eliminate deadlock by informing the operator that a deadlock has occurred and lets the operator deal with the deadlock manually.
   d. We can eliminate deadlock by successively pre-empting some resources from processes and giving these resources to other processes until the deadlock cycle is broken.
   Answer: All of them are true

5. Code:

```
#define N 100
int count = 0;

      void producer(void)
      {
                  int item;
                  while (TRUE) {
                        item = produce_item();
                        if (count == N) sleep();
                        insert_item(item);
                        count = count + 1;
                        if (count == 1) wakeup(consumer);
                  }
      }

      void consumer(void)
      {
                  int item;
                  while (TRUE) {
                        if (count == 0) sleep();
                        item = remove_item();
                        count = count - 1;
                        if (count == N-1) wakeup(producer);
                        consume_item(item);
                  }
      }
```

The procedures insert_item() and consume_item() handle the book-keeping of putting items into the buffer and taking items out of buffer, respectively.
What can this code potentially lead to?

   a. **Race condition**
   b. Starvation
   c. The code is thread-safe. It won't cause any issues.
   d. Circular wait

6. Choose the correct answer based on the following statements regarding shared memory that is accessed by multiple threads.
   a. **Shared memory can run into race conditions if it is not accessed in the correct order by the threads.**
   b. Since the shared memory is isolated between the threads, the main advantage of using shared memory is that it can never run into race conditions.

c. **Shared memory has a serious disadvantage of leaving processes to starve since a single thread can hog the resource by never letting the other threads to access it.**
d. Even though the memory is shared, there need not be any synchronization mechanisms for shared memory because the OS will take care of the thread management on its own.
e. **Shared memory provides an extremely fast way to communicate large or small amounts of data because any data, that is written by one thread to a shared memory region, can be read immediately by any other thread that has the privilege to read from that memory location.**

7. What are the conditions that must be met for a deadlock to occur? Select all the options that are true.
   a. **Mutual Exclusion**
   b. **Hold and wait**
   c. **No preemption**
   d. **Circular wait**
   e. Starvation
   f. Intermittent I/O
   g. Buffer overflow
   h. Bounded buffer

8. What are the different ways of doing IPC? Select all options that are true.
   a. **Signals**
   b. **Interrupts**
   c. **Message passing**
   d. **Shared memory**
   e. **Remote procedure calls**
   f. Parameter passing
   g. loadable kernel modules
   h. Kernel I/O

9. True or False.
   a. A function defined within the monitor can access only those variables which are declared locally within the monitor. **True**
   b. Local variables of a monitor can be accessed only by its local functions. **True**

10. Select the best answer regarding the given code snippet.

```
int temp;

void swap(int* y, int* z)
{
      int local;
      local = temp;
      temp = *y;
      *y = *z;
      *z = temp;
      Temp = local;
}
```

   a. The code is neither thread safe nor re-entrant.
   b. The code is re-entrant but not thread safe.
   c. The code is re-entrant and thread safe.
   d. The code is thread safe but not re-entrant.

11. Select all the options that are true about pipes and sockets.
   a. Sockets in general use a client-server architecture.
   b. Sockets can allow structured stream of bytes to be exchanged between the communicating threads.
   c. Pipes communicate by means of producer-consumer fashion.
   d. Ordinary pipes are bi-directional, which allow two-way communication.
   e. Named pipes require no parent-child relationship.
   f. Named pipes can be used over a network, while ordinary pipes cannot be used.

12. The more deadlocks occur, the lesser the deadlock prevention algorithms should be run because those algorithms can potentially lead to deadlocks. **False**