1. Explain the difference between a hard real-time, a soft real-time and a firm real-time system. Discuss whether the following requirements are hard, firm or soft real-time:

   i. Refreshing the frame of a first person shooting game every 16 milliseconds.
   ii. An antilock braking system (ABS) checking tire every 10 milliseconds.
   iii. Retrieving specific dimensions of a part (i.e., width, height) in an assembly line using computer vision every 500 milliseconds.
   iv. An inkjet printer depositing ink on a paper within 1 milliseconds upon request from hardware.

2. In real-time systems, what are the factors that affect interrupt response times?

3. 2 tasks are being executed in a cyclic coding structure in a RTOS system, as following:

```
for (;;) {
   task_1();
   task_2();
}

void task_1() {
  check_Alarm_1();
   process_A();
   process_B();
}

void task_2() {
  check_Alarm_2();
   process_C();
   process_D();
 }
```

Besides executing their processes, these two tasks check the two alarm lines periodically, and they are supposed to raise the alarm if a signal is received from the corresponding alarm line. The signal in the alarm lines can be as short as 10 milliseconds, and the system is supposed to catch every alarm signal. For each task, checking the alarm input and activating the alarm takes at most 1 milliseconds.

Considering that the worst-case execution times of all processes (A, B, C and D) may take as long as 3 milliseconds each, explain the problem in the current design and solve it using coroutines.

4. Implement a constant size, single reader single writer ring buffer in C++ without locks by implementing the following IRingBuffer interface:

Single reader single writer ring buffer: Two threads can access this buffer concurrently, but only one thread can push, and only one thread can pop.

```
class IRingBuffer {
public:
   const int BUFFERSIZE = 100;
   bool push_back(const int value) = 0; // return false in case of overflow
   bool pop_front(int &value) = 0; // return false in case of underflow
 };
```

5. DMA
   i. Explain DMA by comparing it with polled I/O and interrupt driven I/O
   ii. Explain synchronous and asynchronous DMA and their differences
   iii. Give a step by step example for an asynchronous DMA read operation. Write down the software component types involved in each operation (i.e., user program, operating system, device driver)