

Concurrency and processes

1. Consider the following code, with the assumption that each executes in a separate thread running concurrently:
 - a. Is it guaranteed that someone wins? Explain
 - b. Is it possible for someone to win? Explain

```
i=0
while (i<10)
  i++;
printf("a wins");
```

```
i=0
while (i>-10)
  i--;
printf("b wins");
```

2. In a given operating system its scheduler may perform a context switch 100-1000 times a second. What is a context switch, and what does the OS have to do during one?
 - a. Why is a context switch performed so often?
 - b. Why isn't a context switch performed more often?
3. Consider the following pointless loop written in BASIC that counts upwards indefinitely

```
10 i = i + 1
20 GOTO 10
```

What would happen in this case

4. Describe 3 non-computer activities that could cause deadlock.
 - a. If a computer is in a deadlock state how would it feel to the user? How would you know if it was deadlock and not just a program running a never-ending loop?
 - b. We can have deadlock detection methods and also deadlock avoidance mechanisms. How would a deadlock avoidance mechanism run?
 - c. What are the advantages of a deadlock avoidance mechanism?
 - d. What are the disadvantages of a deadlock avoidance mechanism?