COMP 2432 Operating Systems
Tutorial 12 Solution

1. Andes Trains Solution 1.

There can be *numerous* possible sequences of events (or execution sequences) in these questions. Here are just some examples. Note that there can be *multiple statements* in <CS> and <RS>. You could put more than one copy of statements (3) and (5) in your examples to indicate a longer <CS> and <RS> than a single statement.

(*a*)

Time	$P_{Bolivia}$	P_{Peru}	*flag*
0			false
1	1		
2	2		true
3		1	
4		1	
5	3		
6		1	
7	3		
8	4		false
9	5		
10	1		
11	2		true
12		1	
13	3		
14	3		
15		1	
16	3		
17		1	
18	4		false
19	5		
20	5		
21	1		
22	2		true
23		1	
24	3		
25		1	**blocked**

(*b*)

Time	$P_{Bolivia}$	P_{Peru}	*flag*
0			false
1		1	
2	1		
3	2		true
4		2	true
5		3	
6	3		**crash**
7			
8			
9			
10			
11			
12			
13			
14			
15			
16			
17			
18			
19			
20			
21			
22			
23			
24			
25			

2. Andes Trains Solution 2.

Initially, flag = **false**

$P_{Bolivia}$

```
while (true) do {
1: while (flag == true) do { };
2: < critical section >
3: flag = true;
4: < remainder section >
5:
}
```

P_{Peru}

```
while (true) do {
1: while (flag == false) do { };
2: < critical section >
3: flag = false;
4: < remainder section >
5:
}
```

Time	$P_{Bolivia}$	P_{Peru}	*flag*
0			false
1		1	
2	1		
3		1	
4	2		
5	2		
6	3		true
7		2	
8	4		
9		2	
10	4		
11	1		
12	1		
13		3	false
14		4	
15		1	*Peruvian cannot take a second run until Bolivan takes its second!*
16		1	
17		1	
18		1	
19	2		

3. Andes Trains Solution 3.

Initially, flag[0] = **false** and flag[1] = **false**

$P_{Bolivia}$

```
while (true) do {
1: while (true) do {
2:     flag[0] = true;
3:     if (flag[1] == false) then
4:            break;
5:     else flag[0] = false;
6: }
7: < critical section >
8: flag[0] = false;
9: < remainder section >
}
```

P_{Peru}

```
while (true) do {
1: while (true) do {
2:     flag[1] = true;
3:     if (flag[0] == false) then
4:            break;
5:     else flag[1] = false;
6: }
7: < critical section >
8: flag[1] = false;
9: < remainder section >
}
```

(*a*)

Time	$P_{Bolivia}$	P_{Peru}	*flag*[0]	*flag*[1]
0			false	false
1	1			
2		1		
3		2		true
4		3		
5		4		
6		7		
7	2		true	
8	3			
9	5		false	
10	1			
11		8		false
12		9		
13	2		true	
14	3			
15	4			
16	7			
17	8		false	
18	9			
19		1		
20		2		true
21		3		
22		4		
23		7		
24	9			
25		8		false
26		9		

(*b*)

Time	$P_{Bolivia}$	P_{Peru}	*flag*[0]	*flag*[1]
0			false	false
1	1			
2		1		
3	2		true	
4		2		true
5	3			
6		3		
7	5		false	
8		5		false
9	1			
10		1		
11	2		true	
12		2		true
13	3			
14		3		
15	5		false	
16		5		false
17	1			
18		1		
19	2		true	
20		2		true
21	3			
22		3		
23	5		false	
24		5		false
25	1		***livelock for both trains***	
26		1		

4. Critical Section Problem.

Initially, both *flag*[1] and *flag*[2] are *false* and *turn* = 1.

Program for P_1	Program for P_2

`while (true) do {`
`1: flag[1] = true;`
`2: turn = 1;`
`3: while (flag[2] and turn == 2) do { };`
`4: < critical section >`
`5: flag[1] = false;`
`6: < remainder section >`
`}`
	`while (true) do {`
	`1: flag[2] = true;`
	`2: turn = 2;`
	`3: while (flag[1] and turn == 1) do { };`
	`4: < critical section >`
	`5: flag[2] = false;`
	`6: < remainder section >`
	`}`

There can be ***many*** possible answers. Only one of them is shown here.

ME

Time	P_1	P_2	*flag*[1]	*flag*[2]	*turn*	Time	P_1	P_2	*flag*[1]	*flag*[2]	*turn*
0			false	false	1	8		4			
1	1		true			9	4				
2	2				1	10	**crash**				
3	3					11					
4		1		true		12					
5		2			2	13					
6		3				14					
7		4				15					

PG

Time	P_1	P_2	*flag*[1]	*flag*[2]	*turn*	Time	P_1	P_2	*flag*[1]	*flag*[2]	*turn*
0			false	false	1	13	1		true		
1	1		true			14		1		true	
2		1		true		15		2			2
3	2				1	16		**3**			
4	**3**					17		**4**			
5	**4**					18		5		false	
6	5		false			19		6			
7	6					20	2				1
8		2			2	21		1		true	
9		**3**				22		2			2
10		**4**				23	**3**				
11		5		false		24		**3**			
12		6				25		**4**			

3: ***someone*** wants to enter <CS> / 4: ***someone*** enters <CS>

BW

Time	P_1	P_2	*flag*[1]	*flag*[2]	*turn*	Time	P_1	P_2	*flag*[1]	*flag*[2]	*turn*
0			false	false	1	13	1		true		
1	1		true			14	2				1
2		1		true		15		**3**			
3		2			2	16		**3**			
4	2				1	17	**3**				
5	**3**					18	**4**				
6		**3**				19		**3**			
7	**4**					20	4				
8	4					21	5		false		
9		**3**				22	6				
10		**3**				23	1		true		
11	5		false			24	2				1
12	6					25		**3**	**P_2 starves**		

3: ***I*** want to enter <CS> / 4: ***I*** enter <CS>