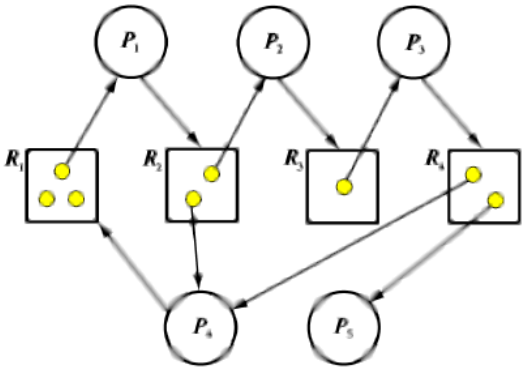
SET 1

Let a system consist of 4 resources which is being shared by 5 processes at time T0. The resources allocated to the various processes is provided in the resource allocation graph. Let the priorities of the processes be P1 -5 , P2- 3, P3 -2, P4-0, P5 – 1.



A) At time T0: Is the system safe? (3 marks)
B) At T1: A request for R1 by P5 is granted. Is the system in deadlock? (3 marks)
C) If the system is in the deadlock state, list the processes that contribute to the deadlock. (2 marks)
D) Which processes will be terminated to recover from deadlock? List the order of termination. (2 marks)

Process P0:	Process P1:	Process 2:
{	{	{
S1: register1=count;	T1: register2=count;	U1: register3=count;
S2: register1=register1 - 5	T2: register2=register2 X 5	U2: register3=register3 + 5
S3: count=register1	T3: count=register2	U3: count=register3
}	}	}

Let the initial value of count be 10. What will be final value of count if the instructions are executed in the following order:
S1; T1; S2; T2;S3;T3;U1; S1; U2;T1; T2;S2;U3 (4 marks).
(list the value of count and the registers after each instruction)

Propose a solution to this problem using hardware atomic function swap. Justify that the solution provides mutual exclusion. (6 marks)

Lamport's bakery algorithm is given below:

Shared variables:

boolean flag choosing[i] initially false;
integer num[i] initially 0

The entry and exit codes for process i are:

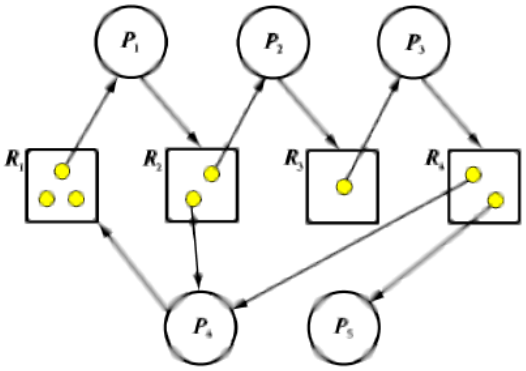
```
entry(i) {
   i.1: choosing[i] := TRUE ;
   i.2: num[i] := max( num[0], num[1], ... , num[N-1] ) + 1 ;
   i.3: choosing[i] := FALSE ;
        for p := 0 to N-1 do  {       // p is local to process i
   i.4:   while choosing[p] do skip ;
   i.5:   while num[p] != 0 and (num[p],p)<(num[i],i) do skip ;
        }
}

exit(i) {
   i.6: num[i] := 0 ;
}
```

Assume a system has five processes P0 – P4. At time T0 P4 is executing in the critical section with num[4] = 10. Assume that the following events happen in the given order:
T1: P0 executes i.1
T2: P0 executes i.2 loses processor
T3: P1 executes i.1
T4: P1 executes i.2
T5: P1 executes i.3 loses processor
T6: P0 executes i.3
T7: P0 executes i.4 loses processor
T8: P2 executes i.1
T9: P2 executes i.2
T10: P2 executes i.3 loses processor
Answer the following questions:
A) What will be the value of num for P0, P1 and P2? (2 marks)
B) Which process will be the first to enter the critical section and why? Assume P0 acquires the processor at T11. (2 marks)
C) Which process will the second to enter the critical section and why? (2 marks)
D) Is mutual exclusion and progress implemented in the algorithm? how? (4 marks)

SET 2

Let a system consist of 4 resources which is being shared by 5 processes at time T0. The resources allocated to the various processes is provided in the resource allocation graph. Let the priorities of the processes be P1 -5 , P2- 3, P3 -2, P4-0, P5 – 1.



A) At time T0: Is the system safe? (3 marks)
B) At T1: P5 releases R4. At T2: A request for R4 by P2 is granted. Is the system in deadlock? (3 marks)
B) If the system is in the deadlock state, list the processes that contribute to the deadlock. (2 marks)
D) Which processes will be terminated to recover from deadlock? List the order of termination. (2 marks)

Process P0:	Process P1:	Process 2:

{
S1: register1=count;
S2: register1=register1 - 5
S3: count=register1
}
	{
	T1: register2=count;
	T2: register2=register2 X 5
	T3: count=register2
	}
		{
		U1: register3=count;
		U2: register3=register3 + 5
		U3: count=register3
		}

Let the initial value of count be 10. What will be final value of count if the instructions are executed in the following order:
T1; S1; T2; S2;S3;T3;U1; S1; U2;T1; T2;S2;U3 (4 marks).
(list the value of count and the registers after each instruction)

Propose a solution to this problem using hardware atomic function test and set. Justify that the solution provides bounded waiting. (6 marks)

Lamport's bakery algorithm is given below:

Shared variables:

boolean flag choosing[i] initially false;
integer num[i] initially 0

The entry and exit codes for process i are:

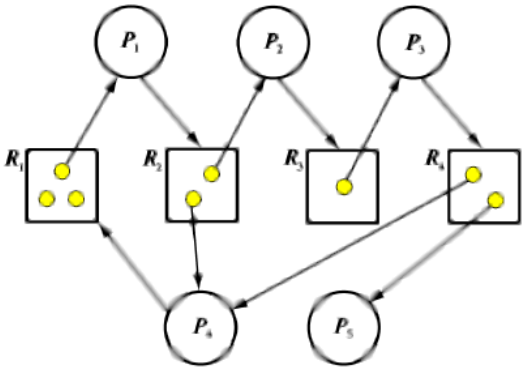
```
entry(i) {
   i.1: choosing[i] := TRUE ;
   i.2: num[i] := max( num[0], num[1], ... , num[N-1] ) + 1 ;
   i.3: choosing[i] := FALSE ;
        for p := 0 to N-1 do  {       // p is local to process i
   i.4:   while choosing[p] do skip ;
   i.5:   while num[p] != 0 and (num[p],p)<(num[i],i) do skip ;
        }
}

exit(i) {
   i.6: num[i] := 0 ;
}
```

Assume a system has five processes P0 – P4. At time T0 P4 is executing in the critical section with num[4] = 10. Assume that the following events happen in the given order:
T1: P1 executes i.1
T2: P1 executes i.2 loses processor
T3: P0 executes i.1
T4: P0 executes i.2
T5: P0 executes i.3 loses processor
T6: P1 executes i.3
T7: P1 executes i.4 loses processor
T8: P2 executes i.1
T9: P2 executes i.2
T10: P2 executes i.3 loses processor
Answer the following questions:
A) What will be the value of num for P0, P1 and P2? (2 marks)
B) Which process will be the first to enter the critical section and why? Assume P0 acquires the processor at T11. (2 marks)
C) Which process will the second to enter the critical section and why? (2 marks)
D) Is mutual exclusion and progress implemented in the algorithm? how? (4 marks)

SET 3

Let a system consist of 4 resources which is being shared by 5 processes at time T0. The resources allocated to the various processes is provided in the resource allocation graph. Let the priorities of the processes be P1 -5 , P2- 3, P3 -2, P4-0, P5 – 1.



A) At time T0: Is the system safe? (3 marks)
B) At T1: P5 releases R4. At T2: A request for R1 by P4 is granted. At T3: A request for R4 by P2 is granted. Is the system in deadlock? (3 marks)
C) If the system is in the deadlock state, list the processes that contribute to the deadlock. (2 marks)
D) Which processes will be terminated to recover from deadlock? List the order of termination. (2 marks)

Process P0:	Process P1:	Process 2:

{
S1: register1=count;
S2: register1=register1 - 5
S3: count=register1
}
	{
	T1: register2=count;
	T2: register2=register2 X 5
	T3: count=register2
	}
		{
		U1: register3=count;
		U2: register3=register3 + 5
		U3: count=register3
		}

Let the initial value of count be 10. What will be final value of count if the instructions are executed in the following order:
U1; S1; T1;U2; S2; T2;S3;T3; S1; T1; T2;S2;U3;S3 (4 marks).
(list the value of count and the registers after each instruction)

Propose a solution to this problem using hardware atomic function swap. Justify that the solution provides mutual exclusion. (6 marks)

Lamport's bakery algorithm is given below:

Shared variables:

boolean flag choosing[i] initially false;
integer num[i] initially 0

The entry and exit codes for process i are:

```
entry(i) {
   i.1: choosing[i] := TRUE ;
   i.2: num[i] := max( num[0], num[1], ... , num[N-1] ) + 1 ;
   i.3: choosing[i] := FALSE ;
        for p := 0 to N-1 do  {       // p is local to process i
   i.4:   while choosing[p] do skip ;
   i.5:   while num[p] != 0 and (num[p],p)<(num[i],i) do skip ;
        }
}

exit(i) {
   i.6: num[i] := 0 ;
}
```

Assume a system has five processes P0 – P4. At time T0 P4 is executing in the critical section with num[4] = 10. Assume that the following events happen in the given order:
T1: P2 executes i.1
T2: P2 executes i.2 loses processor
T3: P1 executes i.1
T4: P1 executes i.2
T5: P1 executes i.3 loses processor
T6: P2 executes i.3
T7: P2 executes i.4 loses processor
T8: P0 executes i.1
T9: P0 executes i.2
T10: P0 executes i.3 loses processor
Answer the following questions:
A) What will be the value of num for P0, P1 and P2? (2 marks)
B) Which process will be the first to enter the critical section and why? Assume P0 acquires the processor at T11. (2 marks)
C) Which process will the second to enter the critical section and why? (2 marks)
D) Is mutual exclusion and progress implemented in the algorithm? how? (4 marks)