1. Consider two functions, P and Q, and the main program creates two threads to execute P and Q as shown in the following code. Note that each statement is labeled and the label is just for convenience.

```
function P();      function Q();
{ a = b + 1; print(a); }   { b = a + 3; print (b); }
main()
{ a = 5;   b = 1;
create a thread to execute P();
create a thread to execute Q(); }
```

List all possible printouts after the execution of the program. You should consider potential process switches at the instruction level.

2. Consider the following program. Find a counter example (execution sequence) that demonstrates that this software solution is incorrect in terms of mutual exclusion violation.

```
var blocked: array[0..1] of boolean;
var turn: 0..1;  // could be 0 or 1

function p (id: integer)
{   repeat
    blocked[id] := true;
    while (turn != id) do
        { while blocked[1–id] do;  // do nothing
            turn := id;
        }
    < critical section >
    blocked[id] := false;
    < remainder code for p>
    until false;
}

main ()
{   blocked[0] := false; blocked[1] := false;
create thread to execute P(0);
create thread to execute P(1);
}
```

3. Consider a sequential program as follows. Assume that we try to execute all statements concurrently to achieve maximal parallelism, but we want to obtain the same output as the sequential execution below. Use (a) locks and (b) semaphores to control the synchronization so that the program can be executed in maximal concurrency while still get the same outcome as the sequential execution.

```
x = y+z;
w = y*z;
u = w–3;
v = x+t;
k = x+w;
```
In the Jurassic Park Amusement center, tourists ride on a tour bus to visit the park. Each bus takes two visitors from the visitor center, drives around the park as long as the passengers wish, and return to the visitor center to drop off the passengers. There are N tour buses on service. If the N buses are all out, then a passenger who wants a ride must wait. If a bus is ready to load but there are no waiting visitors, then the bus waits. A bus does not leave until it loads two passengers. A bus does not load passengers if there is another bus that is only partly filled.

Consider the problem of synchronizing the bus processes and the visitor processes using semaphores. The code for visitor process follows. Three semaphores cust, bus, and busReady are used. Write the code for the bus process that performs the proper operations on the three semaphores. Use extra semaphores as needed. In the bus process, you can use the same pseudo-statements “visitor get in bus”, etc. as in the visitor process. Do not forget to properly initialize the semaphores (including the three given in the code and any new ones you may introduce).

```
var cust: semaphore (:= ?);
    bus: semaphore (:= ?);
    busReady: semaphore (:= ?);

procedure visitor;
begnin
    signal (cust);
    wait (bus);
    visitor gets in bus;
    wait (busReady);
    drive around the park;
    visitor gets off bus;
end;
```

Implement a monitor solution for the bakery problem. Assume that you have N salesmen and customers arrive at arbitrary times. The code for customers and salesmen external to the monitor is given as follows.

```
customer
{ get_service ();
    receive the service
        release_service ();
}
salesman
{ prepare_service();
    provide the service
        complete_service();
}
```

When we implement the code for realizing the monitor, there is one important issue. Monitor can only allow one active thread in it. When signaling a condition variable, we need to decide whether to let the signaler or the signalee to continue. We have discussed how to implement the monitor with the choice of letting the signaler continue. Now, discuss the high-level implementation idea if we choose to let the signalee continue.
7. Consider the bakery problem, but we replace the bakery by the hair salon. There are \( N \) hair dressers working for the salon. The customer can get the hair styling service if a hairdresser is available. The salon also has an entertainment center which has \( M \) seats. A customer can wait in the entertainment center for a hairdresser to become available. After finishing hair styling service, the customer makes payment to the salon.

A salon manager process coordinates the customers and hairdressers. It can admit at most \( N+M \) customers. A customer, when wanting to have the hair styling service, sends a message to the salon manager. The manager “receives” customers from the “custport” and place them in the entertainment center. If there is a waiting customer, the manager “receives” an available hairdresser from the “dresserport” and pairs the customer and the hairdresser for service. If there is no waiting customer, the manager should not receive a hairdresser (hairdressers wait in a room). After the hair styling service finishes, the customer should make payment via the “payport”. The customer and hairdresser processes are given as follows. Give the code for the salon manager process using “Guarded Communication”.

Customer process:

- send (custport, customer);
- take a seat in the entertainment center and wait;
- get service from the hairdresser;
- send (payport, payment);

Hairdresser process:

- repeat select
  - send (dresserport, hairdresser);
  - provide hair styling service;
- until false;

8. Consider a memory system with 1GB space (=1024MB). Currently the memory free list contains \(((128MB, 64MB), (320MB, 8MB), (512MB, 16MB), (768MB, 256MB))\) where each item \((x, y)\) in the list represents the starting location \( x \) and length \( y \) of a free partition. Four jobs \( J_1, J_2, J_3, \) and \( J_4 \) of sizes 14MB, 20MB, 160MB, and 64MB, respectively, are to be loaded to the memory. Consider the first fit allocation policy and answer the following questions.

(a) Consider each job \( J_i \), \( 1 \leq i \leq 4 \). When \( J_i \) is running, what would be the value for the base register and bound register?

(b) What is the free list after loading all four jobs?

(c) Now \( J_4 \) is running and it accesses the logical address 0x04213a0c. Compute the corresponding physical address. Also, check and determine whether this causes access violation.

(d) Now \( J_4 \) is running and it accesses the logical address 0x02f3f12c. Compute the corresponding physical address. Also, check and determine whether this causes access violation.

(e) Answer the questions (a) through (d) for the best fit allocation policy.

9. Consider a 1GB memory system using buddy scheme. The system loads jobs \( J_1, J_2, J_3, J_4, \) and \( J_5 \) of size 40MB, 20MB, 160MB, 64MB, and 100MB, respectively.

(a) Compute the starting addresses of each job.

(b) Give the list of free partitions. For each, include its starting address and size.

(c) What is the total internal fragmentation?
10. Consider a 256MB memory system using simple paging scheme. Each page is of size 16KB. The memory is byte addressable. The page table for Process $P$ is as follows. Process $P$'s address space is only 5 pages. The numbers in the table are all in decimal.

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>11</td>
</tr>
<tr>
<td>1</td>
<td>25</td>
</tr>
<tr>
<td>2</td>
<td>40</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>13</td>
</tr>
</tbody>
</table>

(a) How many bits are required to address the entire memory?
(b) How many bits are required to address the offset within each page?
(c) What is the maximal possible internal fragmentation $P$ can have?
(d) Given a logical address 0x0000ef5b, compute the corresponding physical address. Also, check and determine whether this causes access violation.
(e) Given a logical address 0x00017f5b, compute the corresponding physical address. Also, check and determine whether this causes access violation.

11. Consider a virtual memory system with cache, main memory, and disk. The cache size is 256KB and has an access time 15ns. The cache hit ratio is 90% (which is considered low). The main memory is 256MB and has an access time 150ns. 99.99% of the memory accesses would find the demanded pages in the main memory and 0.01% of them would generate page faults. The disk size is 20GB and has an access time 1ms. Compute the average memory access time.

12. Consider a virtual memory system using 32-bit addressing. The physical memory is of size 4GB. Each page is of size 8KB. Assume that each entry of the page table fits in one word. A two-level page table is used and the root level contains 1K entries. A process $P$ has the following page table.

<table>
<thead>
<tr>
<th>Root level page table</th>
<th>Level-2 page table starting at 0x00000000</th>
<th>Level-2 page table starting at 0x00000100</th>
<th>Level-2 page table starting at 0x00001000</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x00001000</td>
<td>Frame 150</td>
<td>Frame 33</td>
<td>Frame 160</td>
</tr>
<tr>
<td>0x00000100</td>
<td>Frame 21</td>
<td>disk</td>
<td>Frame 512</td>
</tr>
<tr>
<td>null</td>
<td>disk</td>
<td>Frame 120</td>
<td>........</td>
</tr>
<tr>
<td>........</td>
<td>........</td>
<td>........</td>
<td>........</td>
</tr>
</tbody>
</table>

(a) How many bits are required to address the root page table?
(b) How many bits are required to address each second level page table?
(c) Convert the logical address 0000 0000 0100 0000 0101 0000 1111 0101 (binary) to physical address.