Problem A

Consider an intrusion detection system that stores all the source IP addresses observed at the network interface of a router, using a table implemented with a simple vector array. A fingerprinting scheme is adopted. Let \( x = a.b.c.d \) be an IP address expressed with the canonical representation with four decimal digits; the corresponding fingerprint is \( F(x) = (a + b + c + d) \mod 16 \).

1. Consider the sequence of operations in the above table. For each operation, show the content of table \( T \) and the probability of false positive after the operation. Assume the table initially empty: \( T = \{ \} \).

<table>
<thead>
<tr>
<th>Time</th>
<th>Operation</th>
<th>Table ( T )</th>
<th>Prob. false positive</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>insert(1.2.3.4)</td>
<td>{}</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>insert(2.3.4.5)</td>
<td>{}</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>insert(2.3.4.1)</td>
<td>{}</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>insert(3.4.5.2)</td>
<td>{}</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>delete(1.2.3.4)</td>
<td>{}</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>delete(2.3.4.1)</td>
<td>{}</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>delete(3.4.5.2)</td>
<td>{}</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>delete(2.3.4.5)</td>
<td>{}</td>
<td></td>
</tr>
</tbody>
</table>

2. In the above sequence of operations, add at some time an operation for which a false positive event occurs.

3. Assume that \( k \) IP addresses are stored exploiting fingerprints of \( b \) bits. What is the total required storage in terms of bits? What is the final probability of false positives?

4. Assume that \( k \) IP addresses are stored directly in the table without fingerprinting. Compare this solution with the one adopting fingerprinting in terms of storage requirement and probability of false positives.

5. Show the pseudo-code to insert, delete and search an element in a table exploiting fingerprinting.
Problem B

Design a rearrangeable switch of size 10,000 × 10,000 using only modules of size 10 × 10, with the aim of minimizing the number of modules.

1. Describe the architecture.
2. Compute the total number of required modules.
3. Describe the configuration algorithm.
4. Write the formula to compute the minimum theoretical number of 10 × 10 modules. Note that the formula must be compatible with a standard scientific calculator. Recall the Stirling approximation: $N! \approx \sqrt{2\pi N} (N/e)^N$. 
Problem C

Consider a slotted input queued switch, of size $N \times M$ and with Virtual Output Queueing. The switch is fed by variable-size packets.

1. Describe in pseudo-code a scheduling algorithm working in cell-mode.

2. Describe in pseudo-code a scheduling algorithm working in packet-mode.

3. Compare the performance of the cell-mode algorithm and the packet-mode algorithm in terms of throughput and delay.
Hints for the solution

Problem A

1. The evolution of the table is reported below. Note that multiple copies of the same fingerprint must be stored to support deletion.

<table>
<thead>
<tr>
<th>Time</th>
<th>Operation</th>
<th>Table $T$</th>
<th>Prob. false positive</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>insert(1.2.3.4)</td>
<td>{10}</td>
<td>1/16</td>
</tr>
<tr>
<td>2</td>
<td>insert(2.3.4.5)</td>
<td>{10, 14}</td>
<td>1/8</td>
</tr>
<tr>
<td>3</td>
<td>insert(2.3.4.1)</td>
<td>{10, 10, 14}</td>
<td>1/8</td>
</tr>
<tr>
<td>4</td>
<td>insert(3.4.5.2)</td>
<td>{10, 10, 14, 14}</td>
<td>1/8</td>
</tr>
<tr>
<td>5</td>
<td>delete(1.2.3.4)</td>
<td>{10, 14, 14}</td>
<td>1/8</td>
</tr>
<tr>
<td>6</td>
<td>delete(2.3.4.1)</td>
<td>{14, 14}</td>
<td>1/16</td>
</tr>
<tr>
<td>7</td>
<td>delete(3.4.5.2)</td>
<td>{14}</td>
<td>1/16</td>
</tr>
<tr>
<td>8</td>
<td>delete(2.3.4.5)</td>
<td>{}</td>
<td>0</td>
</tr>
</tbody>
</table>

2. E.g., Search(2.1.4.3) at time 1.5 or Search(1.2.3.4) at time 5.5 would lead to a false positive event.

3. The total storage is $bk$ bits. The final probability of false positives is

$$1 - \left( 1 - \frac{1}{2^b} \right)^k$$

4. Without fingerprinting, each IP address is represented with 32 bits. Thus the storage requirement is $32k$ bits in total. Note that this data structure avoids false positive events, but requires a larger storage which is $32/k$-times the one with fingerprinting (assuming $b < 32$).

5. Assume that the size of the vector array $T$ is $n$. Assume that all the entries in $T$ have been initialized to $-1$ as default value.

```plaintext
function INSERT(x)
    for $i = 1 \rightarrow n$
        if $T[i] = -1$
            $T[i] = F(x)$
            return OK
        end if
    end for
    return ERROR_FULL_TABLE
end function

function FIND-ELEMENT(x)
    for $i = 1 \rightarrow n$
        if $T[i] = F(x)$
            return FOUND
        end if
    end for
    return NOT_FOUND
end function

function DELETE-ELEMENT(x)
    for $i = 1 \rightarrow n$
        if $T[i] = F(x)$
            $T[i] = -1$
            return OK
        end if
    end for
    return ERROR_NOT_FOUND
end function
```
Problem B

Exploiting a standard recursive construction:

\[ C_{10000} = 2000C_{10} + 10C_{1000} = (2000 + 10 \times 500)C_{10} = 7000C_{10} \]

being

\[ C_{1000} = 200C_{10} + 10C_{100} = (200 + 300)C_{10} = 500C_{10} \]

being

\[ C_{100} = 30C_{10} \]

The minimum number of modules can be computed as:

\[ \hat{C}_{10} = \log_{10}(10000!) = \frac{\log(10000!)}{\log(10!)} \approx \frac{0.5 \log(10000) + 10000 \log(10000)}{0.5 \log(10) + 10 \log(10)} \]

Using a calculator:

\[ \hat{C}_{10} = 3809 \]

which implies that the proposed solution is only \((7000 - 3809)/3809 = 83\%\) larger than the lower bound.

Problem C

The solution is provided in ex. 88.