1. (String Matching; 32 points)

(a) (Rabin-Karp algorithm; 8 points) In the Rabin-Karp algorithm, suppose we are searching the text for 2 digit strings. Suppose the hash function for a 2 digit number $n$ is $val(n) = n \mod 7$. So, we have

- **text**: 2 4 1 5 7
- **val(n)**: 3 6 1 1

Construct a string of 8 digits, with as many different digits as possible, in which $val(n) = 5$ for every 2-digit contiguous substring. Avoid 3 3 3 3 3 3 3 3.

(b) (8 points) Let $(ab)^n a$, for $n \geq 1$, denote the string obtained by $n$-fold repetition of $ab$ followed by $a$. Thus, $(ab)^2 a$ is the string $ababa$. Show that $c^i((ab)^n a)$, where $1 \leq i \leq n$, is $(ab)^{n-i} a$.

(c) (6 points) The algorithm for core computation includes the following code fragment; see notes on “String Matching”, Page 10.

```c
if p[u] = p[v]
   then c(v') := u'
   else c(v') := \epsilon
endif;
```

Is it possible that $u = \epsilon$ and $c(v') \neq \epsilon$ after execution of this portion of the program? Show a small example to support your claim.

(d) (KMP Algorithm; 10 points) Apply the KMP algorithm on the following pattern and text. Show only the different values of $l$ (see page 6 of the notes on “String Matching”).

<table>
<thead>
<tr>
<th>index</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
</tr>
</thead>
<tbody>
<tr>
<td>text</td>
<td>a</td>
<td>b</td>
<td>a</td>
<td>b</td>
<td>a</td>
<td>a</td>
<td>b</td>
<td>b</td>
<td>a</td>
<td>b</td>
<td>a</td>
<td>b</td>
</tr>
</tbody>
</table>
| pattern | a | b | a | b | a | b | a | b

2. (Data Parallel Programming; 18 points)

(a) (8 points) Consider the function $f$ on integers that returns a powerlist, defined as follows.

- $f(0) = (0 1)$
- $f(n + 1) = (f n) \mid \text{rev}(f n)$
Prove that for \( n \geq 1, f(n + 1) = (f n) \cdot (f n) \). Use the following definition of \( \text{rev} \).

\[
\begin{align*}
\text{rev}(x) &= \langle x \rangle \\
\text{rev}(p \cdot q) &= (\text{rev } q) \cdot (\text{rev } p)
\end{align*}
\]

(b) (10 points) Let \( u \) and \( v \) be powerlists of equal length whose elements are \( n \)-bit binary strings (\( n \) is the same for both \( u \) and \( v \)). Suppose \( u \sim v \) denotes that the corresponding elements of \( u \) and \( v \) are apart by Hamming distance 1. For example, with \( n = 2 \), let

\[
\begin{align*}
x &= 00 \ 01 \ 11 \ 10 \\
y &= 10 \ 11 \ 01 \ 00 \\
z &= 11 \ 10 \ 00 \ 01
\end{align*}
\]

Then, \( x \sim y \), \( y \sim z \), but \( x \not\sim z \).

i. (2 points) Simplify \( (u \bowtie v) \sim (r \bowtie s) \). No proof needed.

ii. (4 points) Define a boolean function \( \text{gr} \) where \( \text{gr } u \) holds iff each pair of adjacent elements of \( u \) (counting wrap-around) are apart by Hamming distance 1. So, \( \text{gr } x \), \( \text{gr } y \) and \( \text{gr } z \) hold. You may use any permutation function defined in the notes.

iii. (4 points) Using your definition of \( \text{gr} \), express \( \text{gr} (u \bowtie v) \) in simpler terms.