- Start as early as possible, and contact the instructor if you get stuck.
- See the course outline for details about the course's grading policy and rules on collaboration.
- Submit your completed solutions to **Crowdmark**.

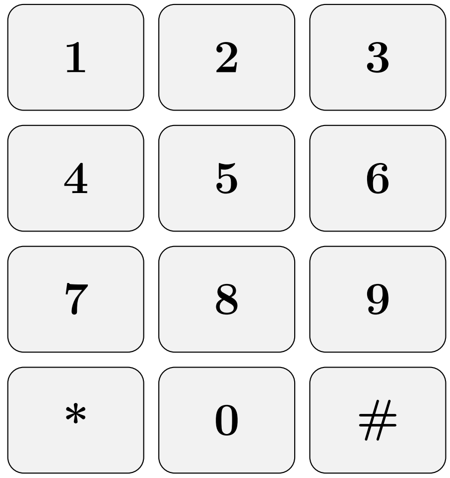
1. **Practice Constructing Automata [9 marks]**
   Let $\Sigma = \{\texttt{0}, \texttt{1}, \texttt{2}\}$. Draw an automaton (DFA, NFA or ε-NFA–your choice) for each of the following languages.
   (a) $L = \{x \in \Sigma^* \mid x$ does not contain 00, 11, or 22 as a substring$\} \subseteq \Sigma^*$.
   (b) $L = \{x \in \Sigma^* \mid$ the first and last symbol are different $(x_1 \neq x_n)\}$.
   (c) $L = \{x \in \Sigma^* \mid$ the last non-0 symbol in x is a 2$\}$.
2. **Unary Regular Languages [8 marks]**
   A *unary language* is any language over an alphabet with one symbol, $\Sigma = \{\texttt{1}\}$. Consider the following claim.

   Claim 1. *Define unary languages $M_{a,b} = \{\texttt{1}^{a+bi} : i \geq 0\}$ for all non-negative integers a, b. For any unary regular language $L \subseteq \{\texttt{1}\}^*$ (that is, a language that is both unary and regular), it can be written as a finite union*

   $$L = \bigcup_{i=1}^{m} M_{a_i,b_i}$$

   where m, $a_1, \ldots, a_m \geq 0$, and $b_1, \ldots, b_m \geq 1$ depend on L.

   (a) Unfortunately, there is a small bug in Claim 1 as written above. Produce a counterexample: a unary regular language that is not of the form in Claim 1. Briefly justify your counterexample.
   (b) Make a small change to Claim 1 to give a correct characterization of all unary regular languages. Prove that your characterization is correct.
3. **Knight Dialer [9 marks]**
   In a classic interview problem, you are asked to count phone numbers such that each digit is a knight move away from the previous digit (i.e., two steps in one direction and one step in an orthogonal direction) when typed on a pad like the one below. **We also allow repetitions, so adjacent digits can be the same**. Let $\textsc{Knight} \subseteq \Sigma^*$ be the language of all such phone numbers (of any length). E.g., ε, 18816, and 555 are all in $\textsc{Knight}$, but 411 and 8675309 are not.



(a) Demonstrate that KNIGHT is a regular language by providing a DFA, and briefly justifying that it recognizes KNIGHT.

(b) Explain how knowing a DFA for a regular language can help count the number of length-n strings. Apply this to your automaton to count the strings in KNIGHT of length $n = 7$.

Hint: If we replace all transition labels with $\bot$, then the DFA becomes an NFA which accepts $\bot^n$ if and only if some length n string was originally accepted.

4. **Pumping Lemma [8 marks]**
Let $\Sigma = \{0, 1\}$ and let $L \subseteq \Sigma^*$ be the language

$$L = \{uv \mid u, v \in \Sigma^*, u = \overline{v}^R\},$$

where $\cdot^R$ denotes string reversal and $\overline{v}$ is bitwise complement, e.g., $\overline{\texttt{01}} = \texttt{10}$. Prove that L is not regular using the pumping lemma.

5. **Minimization and Reachability [6 marks]**
In class, we saw that we can construct the intersection of two DFAs $M_A = (Q_A, \Sigma, \delta_A, q_A, F_A)$, $M_B = (Q_B, \Sigma, \delta_B, q_B, F_B)$ with a new automaton $M_{A \cap B}$ (the "product construction") using states $Q_A \times Q_B$. We also saw that $M_{A \cap B}$ is not necessarily minimal. For example,

$$A = \{x \in \{0, 1\}^* \mid x \text{ starts and ends with } 0\},$$
$$B = \{x \in \{0, 1\}^* \mid x \text{ starts and ends with } 1\},$$

has an accepting state but it is not reachable (clearly, since $A \cap B = \emptyset$).
For this problem, your job is to pick an alphabet Σ and find minimal DFAs M_A and M_B such that every state (in $Q_A \times Q_B$) is reachable, but $M_{A \cap B}$ is still not minimal.