Let $M$ be the TM with start state $s_0$, accept state $s_1$ and rules

\[
\langle s_0, 0, s_0, 1, R \rangle \quad \langle s_0, 1, s_2, 0, L \rangle \\
\langle s_2, 1, s_3, 1, L \rangle \quad \langle s_3, 1, s_2, 1, L \rangle \\
\langle s_2, B, s_1, 1, P \rangle \quad \langle s_3, B, s_2, R \rangle
\]

Write down the successive configurations of $M$ on inputs

(i) 0, (ii) 001, (iii) 01.

In each case say whether $M$ halts and accepts, halts and does not accept, or does not halt.

2 Describe TM’s accepting each of the following languages, as subsets of $\{0, 1, c\}^*$:

(i) $\{0, 1, c\}^*$.

(ii) $\emptyset$

(iii) $\{\sigma \in \{0, 1, c\}^* | \sigma \text{ contains exactly one } c\}$.

(iv) $\{011\}$.

(v) $\{\sigma c^n \sigma | \sigma \in \{0, 1\}^*, n > 0\}$ where $c^n$ is the word consisting of $n$ copies of $c$.

In each case give the start state, accept state and set of rules of your TM and include a brief description of how it is supposed to work.

3 Let $M$ be a TM with, amongst others, states $q_1, q_2, \ldots, q_k$. Describe a TM $M'$ such that for any input $\sigma \in A^*$,

$M'$ accepts $\sigma \iff M$ halts on $\sigma$ in one of the states $q_1, q_2, \ldots, q_k$.

[In particular then allowing multiple accept states in the definition of a TM does not give any real strengthening.]

4 Show that if $L \subseteq A^*$ is recursive then so is

$\{\sigma \in A^* | \sigma R \in L\}$

where $\sigma R$ means $\sigma$ written backwards.
5 Prove that every finite language is recursive. [Hint: If \( L \subseteq A^* \) is finite then there is an integer \( n \) such that \(|w| < n\) for all \( w \in L \).]

6 Describe a simple 2TM to accept
\[
\{ \sigma \tau \in \{0, 1, c\}^* \mid \sigma \in \{0, 1\}^* \text{ and } |\sigma| = |\tau| \} \subseteq \{0, 1, c\}^*.
\]

7 Given a TM \( M \) describe (briefly) a TM \( M' \) such that for \( \sigma \in A^* \),
   
   (i) if \( M \) accepts \( \sigma \) then \( M' \) accepts \( \sigma \); and
   
   (ii) if \( M \) does not accept \( \sigma \) then \( M' \) does not halt on \( \sigma \).

Hence show that:
   
   (a) \( \{ \sigma \in \{0, 1\}^* \mid \sigma \text{ codes a TM which does not halt on input } \sigma \} \) is not recursive.
   
   (b) \( \{ \sigma \in \{0, 1\}^* \mid \sigma \text{ codes a TM which halts on input } \sigma \} \) is not recursive.

   [This is commonly referred to as the Halting Problem.]

8 As usual let \( U \) be a 1-tape Universal TM. By appealing to the Church-Turing Thesis, or otherwise, show that
\[
\{ \sigma \in \{0, 1\}^* \mid U \text{ accepts } \sigma \}
\]
is not recursive.

9 Let \( P \) be a non-empty proper subset of the set of all codes for TM’s such that if \( \sigma_M, \sigma_N \) code TM’s \( M, N \) respectively and \( M \) and \( N \) are equivalent on \( \{0, 1\}^* \) then
\[
\sigma_M \in P \iff \sigma_N \in P.
\]

Using the Church-Turing Thesis if you wish, show that \( P \) is undecidable (i.e. not recursive). [Harder than your average question.]

   [This fact is essentially Rice’s Theorem (although many textbooks state the theorem in terms recursive functions rather than Turing machines). It is a very powerful tool for proving that languages are not recursive.]

10 By (briefly) describing a suitable 2TM, or otherwise, show that the function \( f : \{0, 1\}^* \to \{0, 1, c\}^* \) given by
\[
f(\sigma) = \sigma c \sigma
\]
is recursive.

11 By describing a suitable 3TM, or otherwise, show that the addition function on \( \langle n, m \rangle \mapsto n + m \) is recursive.
12 By describing a suitable 2TM, or otherwise, show that the function 
\( f : \{0, 1\}^* \rightarrow \mathbb{N} \) given by 
\[ f(\sigma) = |\sigma|, \]
where as usual \(|\sigma|\) is the length of \(\sigma\) (as a binary number) is recursive.

13 By describing a suitable multtape TM show that the subset 
\[ \{(n, m) \in \mathbb{N} \mid n \text{ (exactly) divides } m\} \]
of \(\mathbb{N}^2\) is recursive.

14 Prove that if \( f : \mathcal{A}_1^* \rightarrow \mathcal{A}_2^* \), then \( f \) is recursive if and only if the language 
\[ \{ \sigma \hat{\tau} \mid \sigma \in \mathcal{A}_1^*, \tau \in \mathcal{A}_2^*, f(\sigma) = \tau \} \subseteq (\mathcal{A}_1 \cup \mathcal{A}_2 \cup \{\hat{\tau}\})^* \]
is recursive.

15 Exhibit a deterministic TM with 3 or fewer rules which on the empty input \(\epsilon\) writes onto the tape a word of length at least 4, and then halts.

16 Give the rules for a (1-tape) NDTM to accept 
\[ \{ \sigma \tau \mid \sigma, \tau \in \{0, 1\}^*, \sigma = \sigma^R, \ |\sigma| > 1 \}. \]

17 By describing a suitable multitape NDTM show that if \( L \subseteq \mathcal{A}^* \) is recursive then so is 
\[ \{ \sigma \tau \in \mathcal{A}^* \mid \sigma, \tau \in L \}. \]

18 By constructing a suitable multitape NDTM and assuming that multiplication (on \(\mathbb{N}\)) is recursive show that 
\[ \{ n \in \mathbb{N} \mid n \text{ is not prime} \} \]
and 
\[ \{ n \in \mathbb{N} \mid n \text{ is prime} \} \]
are recursive.

19 Let \( M \) be a NDTM. Show that \( M \) halts on input \( \sigma \in \mathcal{A}^* \) (i.e. there is a \( k \in \mathbb{N} \) such that no computation of \( M \) on \(\sigma\) has more than \( k \) steps/configurations) if and only if there is no infinite computation of \( M \) on \(\sigma\). \(\text{[Harder than your average question. Even if you manage to solve it yourself, see the solution for some remarks on the proof method.]}\)

20 A language \( L \) is called \textit{recursively enumerable} if there exists a (not necessary halting!) TM which, on input \(\sigma\), halts in the accept state if and only if \(\sigma \in L\). Using the Church-Turing thesis, prove that \( L \subseteq \mathcal{A}^* \) is recursive if and only if both \( L \) and \( \mathcal{A}^* \setminus L \) are recursively enumerable.