Answer one question. Each question carries 100 marks. Students are allowed to use course notes, books and calculators.

1. (a) Consider the language \( L_\Sigma = \{ww^Rw \mid w \in \Sigma^* \} \). Show that \( L_{\{a\}} \) is a regular language. Prove that \( L_{\{a,b\}} \) is not a context free language.

(b) A visit-accept automaton is a normal non-deterministic finite-state automaton, with an initial state, labelled transitions and final states which are called accept states. The only difference from normal finite-state automata, is that visit-accept automaton accepts an input string if, starting in its initial state, it can follow transitions labelled by symbols in the input, passing through an accept state somewhere along the transitions.

i. Formalise the notion of a visit-accept automaton.

ii. Formalise the configuration of these automata. 

**Hint:** You must also remember whether an accept state has already been visited.

iii. Define \( C_0 \), the initial configuration, and the set \( C_F \), the set of final configurations.

iv. Define \( C \xrightarrow{a} C' \), meaning that configuration \( C' \) can be reached from configuration \( C \) reading symbol (not string) \( a \).

v. Define \( C \xrightarrow{s} C' \), meaning that configuration \( C' \) can be reached from configuration \( C \) with string \( s \).
vi. Hence or otherwise define the language accepted by a visit-accept automaton.

vii. Give a construction to show that all regular languages can be accepted by visit-accept automata.

**Hint:** Show that a non-deterministic automaton can be converted into one with no outgoing transitions from the final states.

2. (a) A go-back automaton is a normal non-deterministic finite-state automaton, except that it also has labelled transitions with no destination state. When such a transition is taken, the automaton goes back to the previously visited state. Thus, in the following example, taking the $c$ transition from state $C$ would return the automaton to state $A$ or $B$, depending on where it came from. We assume that the initial state has no such transitions.

![Diagram of a go-back automaton]

As in the case of normal automata, a string is accepted if one can follow transitions from the initial state, leading to a final state.

i. Formalise the notion of a go-back automaton.

ii. Formalise the *configuration* of these automata.

**Hint:** You must also remember the list of previously visited states.

iii. Define $C_0$, the initial configuration, and the set $C_F$, the set of final configurations.

iv. Define $C \xrightarrow{a} C'$, meaning that configuration $C'$ can be reached from configuration $C$ reading symbol (not string) $a$. 
v. Define $C \xrightarrow{s} C'$, meaning that configuration $C'$ can be reached from configuration $C$ with string $s$.

vi. Hence or otherwise define the language accepted by a go-back automaton.

(b) Consider the following problem:

<table>
<thead>
<tr>
<th>Reversed Substring:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Given a Turing Machine $M$, input $i$ and length $n$, is there a string $s$ of length $n$ or longer such that both $s$ and its reverse appear on the tape of $M$ when started with input $i$?</td>
</tr>
</tbody>
</table>

Show that Reversed Substring is undecidable.

**Hint:** Show that a Turing Machine can be converted into one that writes a particular string and its reverse only upon termination and then use reducibility.