1. Give a brief explain for the following terms in the context of Turing Machines: state, input tape, move, halt, halting problem, recognise (a language), compute (a function).

2. Outline an implementation-level description of a Turing machine that recognises the language $(10)^*$. State any assumptions you make.

3. Write a state machine version of your Turing machine to recognise the language $(10)^*$ (as in the previous question). Show all the moves of this machine.

4. Outline an implementation-level description of a Turing machine that computes the function $f(n) = n - 2$ where a natural number $n$ is represented in unary.

5. Outline an implementation-level description of a Turing machine that acts as a “doubler”. For Input: A string of 1s of length $n$ and for Output: A string of 1s of length $2n$. This machine calculates the function $f(n) = n + n$ where a natural number $n$ is represented in unary (with 0 being represented as an empty sequence of 1s).

   (If on the other hand we regard the input as being in unary, but with 0 represented using a single 1, then this TM can be regarded as calculating the function $f(n) = 2(n - 1) + 1$. For example, three 1s will represent 2; doubled, we get six 1s, which represents 5; and $2(3 - 1) + 1 = 5$.)

6. Outline an implementation-level description of a Turing machine that computes the function $f(x, y) = \max(x, y)$ where $x$ and $y$ are represented in unary number notation separated by a separator symbol $\ast$. The machine starts at the leftmost non-blank cell. It should leave the
tape containing $z$ in unary, where $z$ is the maximum of $x$ and $y$. Do not use any symbols other than 1, the separator $*$ and blank at any time.