Mid I: 2018 **Theory of Computation** **Total: 30 Marks**

Name and Roll No.: ____________________

1. Recall that $L_u = \{(M, x) : M$ is a Turing machine, x is a string and M accepts $x\}$. Suppose L is a language such that $L_u \leq_m L$. Can we say that $\overline{L}$ (complement of L) is not recursively enumerable? Justify your answer. $\boxed{3}$

   Soln: $L_u \leq_m L \implies \overline{L_u} \leq_m \overline{L}$ (why?). Since $\overline{L_u}$ is not recursively enumerable, it follows that $\overline{L}$ is not recursively enumerable.

2. Let $L_H = \{M : M$ is a Turing machine that halts on all inputs $\}$. Suppose A is a reduction algorithm from L_u to L_H. Suppose a machine string pair (M, x) is given as input to A, what is the property expected of $A(M, x)$, the output of A on the input (M, x)? $\boxed{3}$

   Soln: On input (M, x), the reduction algorithm should produce a machine $M'(= A(M, x))$ as output such that M accepts x if and only if M' halts on every input.

3. Does the fact that L_H is undecidable follow from Rice Theorem? If yes explain how. If not explain why not. $\boxed{3}$

   Soln: L_H is not a property of Turing acceptable languages, but is a property of machines. (A machine may not accept all strings on which it halts - so the property that a machine halts on all inputs does not give any clue on what is the language accepted by it, except that the language is decidable. However, a decidable language can be accepted by infinitely many machines that do not halt.) Hence, Rice theorem gives no conclusion about the decidability of L_H.

4. Show that L_H is not recursively enumerable. $\boxed{6}$

   Soln: Consider the reduction algorithm A that in input (M, x) outputs a machine M' which on its input y, behaves as below:

```
$M'(y) {
   Using UTM, simulate the machine $M$ on input $x$ for $|y|$ steps.
   If $M$ rejects $x$ within $|y|$ steps, then accept $y$.
   If $M$ does not halt in $|y|$ steps then accept $y$.
   If $M$ accepts $x$ within $|y|$ steps, then loop for ever.
}
```

   Now, if M accepts x, then M must accept x in some k steps and M' will go into an infinite loop on all inputs y whose length exceeds k. Thus M' does not halt on all inputs. Otherwise, if M does not accept x, then M' always accepts y, irrespective of what y is. Thus, if $(M, x) \notin L_u$, then $M' \in L_H$ and if $(M, x) \in L_u$, then $M' \notin L_H$. That is, $\overline{L_u} \leq_m L_H$. Since $\overline{L_u}$ is not recursively enumerable, so is L_H.