35 points total

1. Write out the full program of a Turing machine which decides the language $L$ of all binary string which contain an equal number of 0's and 1's. 5 points

   Answer: The idea is, for a given binary input string, we look for an unmarked 0, and we mark such a 0 and match that 0 with an unmarked 1 which we then also mark. When we run out of 0’s, we check if there are remaining unmarked 1’s, and if there are none we accept the string.

   So for a given string, do the following:

   1. Sweep the tape from left to right to find the first unmarked 0.

      If no such symbol exists, **jump to line 4**.

   2. Mark the 0 found in step 1 and move the head to the origin of the tape.

      Otherwise, move the head left back to the origin of the tape.


      If this search fails, reject.

      Otherwise, mark it, move left to the origin of the tape, and **jump to line 1**.

   4. Go to the tape origin. Scan for any unmarked 1’s.

      If no such 1 exists, accept. Otherwise, reject.

2. Describe a TM which accepts the language $L = \{ wCy \mid w \text{ is a binary positive integer, C is just the letter C, and } y \text{ is the binary integer whose value is } 2w \}$. 5 points

   Here I mean informally describe how the TM works. You need not give the full program or diagram.

   We are looking for something like the description of the TM in examples 3.11 or 3.12 of the book on pages 146 and 147.

   Note: A binary positive integer is a binary string whose first bit is a 1.

   Answer: Note that $wCy \in L$ means that $y = w0$. (That is, the string $w$ with a 0 appended at the right.)

   Now given an input string $S$, our TM needs to:

   1. Check if $S$ is of the form $wCy$ where $w$ and $y$ are binary integers (that is binary string without any leading 0’s.) If not, reject.

   2. Check if the first $|w|$ bits of $y$ is equal to $w$. If not, reject.
3. Check if the last bit of y is 0. If yes accept, If not, reject.

3. Explain why the collection of all decidable languages is closed under intersection. 5 points
Use the TM definition of decidable here. So you need to show the if L and J are two decidable languages then L ∩ J is also decidable.

**Answer:**
Let L₁ and L₂ be two decidable languages. Let M₁ and M₂ be TM’s which are their deciders. We need to build a decider that for a given string x, it accepts only if x ∈ L₁ and x ∈ L₂.

M = “On input x:

1. Run M₁ on x. If it rejects, reject and halt.
2. Run M₂ on x, and do whatever M₂ does.”

4. Write a TM which decides the set of binary strings which when interpreted as binary integers are evenly divisible by 8. 5 points
Give the full TM program or draw a state diagram. Hint: This is easier than you might think.

**Brief Answer:** Observe that numbers divisible by 8, in binary notation end with three zeroes. Now, construct a machine that checks if the given binary string ends with three zeroes. If so it accepts, else it rejects. Constructing such a machine is easy. (See problem 2 for a similar example.)

5. Show that a set of natural numbers is decidable if and only if it can be enumerated in increasing order. 10 points

**Answer:** There are two directions to prove here.

1. Assume a set S of natural numbers is decidable, prove that S can be enumerated in increasing order.

**Proof:**
Here is an algorithm which enumerates S in increasing order. The algorithms never halts and it consider all the natural number in increasing order.

Do the following two steps for n = 1,2,3,4.....

i. Since S is decidable there is an algorithms A which decides it. Run A(n). If A(n) accepts then output n, if A(n) rejects then no number if output.

ii. Let n = n+1.

2. Assume a set S of natural numbers can be enumerated in increasing order prove that S is decidable.

**Proof:**
Let E be an algorithm which enumerates S in increasing order. We assume that if S is finite then S eventually halts after enumerating all of the elements in S.
Here is an algorithm $A(n)$ which decides $S$.

On input $n \in \mathbb{N}$, run $E$.
As $E$ runs, if it ever outputs the number $n$ then $A(n)$ halts and accepts $n$.
If $E$ ever outputs a number larger than $n$ the then $A(n)$ halts and rejects $n$.
If $E$ ever halts before outputting $n$ or a number bigger than $n$, then $A(n)$ halts and rejects $n$.

We claim that $A$ decides $S$.
This true because $A$ on any input $n$ $A(n)$ halts.
If $n$ is in $S$ then $A(n)$ accepts and if $n$ is not in $S$ then $A(n)$ rejects.

6. Show that a set $S$ of natural numbers is recognizable and if its complement $\mathbb{N} - S$ is recognizable then $S$ is decidable. 5 points

Ans: We assume a set $S$ of natural numbers is recognizable, and also that the $\mathbb{N} - S$ of natural numbers is recognizable.
Say $S$ is recognized by some algorithm $A$ and $\mathbb{N} - S$ is recognized by some algorithm $B$.
We present an algorithm $C$ to decide $S$.
$C$ on input $n$ works as follows:
Run $A(n)$ and also $B(n)$ interweaving the two computations. That is, do one step of $A$ and one step of $B$, then another step of $A$ and another step of $B$,.... If $A(n)$ ever accepts then $C$ accepts, and if $B(n)$ ever accepts then $C$ rejects.
Now as $B$ recognizes the complement of $A$, as $C$ computes on $n$, either $A(n)$ or $B(n)$ accepts (but not both).
So in its computation, $C(n)$ eventually halts on any input and it accepts exactly $S$. So $S$ is decidable.