1. Let $L$ be a language which is infinite and recognizable. Show that there is a subset $K$ of $L$ which is infinite and decidable.

Answer: Since $L$ is recognizable it is also enumerable.  
So let $l_1, l_2, l_3, \ldots$ be a computable enumeration of $L$. (Note: This enumeration lists every element in $L$, it is in arbitrary order and may contain repeated elements of $L$.)

We give an enumeration of an infinite subset of $L$ which is in increasing order. By the problem in an earlier homework this is sufficient as a set which is enumerated in increasing order is decidable.

Here is the enumeration: $e_1, e_2, e_3, \ldots$ Let $e_1 = l_1$, $e_2$ = the first element in the enumeration of $L$ which is $> e_1$, ....

In general, for any $j > 1$ we let $e_j = $ the first element in the enumeration of $L$ which is $> e_{j-1}$.

It is easy to see this enumeration exists and has the needed properties of being in increasing order and hence the set of all the elements which are enumerated is exactly $L$ and is decidable.

2. True or false: Any finite set $F$ of Boolean strings is decidable.

Answer: This is true. First, as we have mentioned, though you should think this through a bit, the set containing any single finite string is decidable. Second, as shown previously in a homework problem, the union of two decidable sets is decidable. It follows from this that any FINITE union of decidable sets is decidable. (Question: Is this true for an infinite union of decidable sets ??)

Now, as $F$ is finite, it is the union of the finite number of decidable sets each of which is a single element of $F$. So $F$ is decidable.

3. Give a 1-1 and onto correspondence between $N$ and $J$. ($J$ is the set of all integers, positive, negative and 0.)

Answer: Just let 1 correspond to 0, let any odd number $2n+1$ bigger than 1 (so here $n \geq 1$) correspond to $n$, and let $2n$ correspond to $-n$.

4. True or false: The set of all Turing machines (as defined in the textbook) is countable. Explain your reasoning.

Answer: This is true, and actually was done in class. In brief, any subset of a countable set is countable. We can enumerate every finite string over the alphabet of the Turing machines (done in class) and so among all these finite strings are the strings which represent all of the Turing machines. SO the set of Turing machines, being a subset of the set of all finite strings over the alphabet of the Turing machines, is countable.
5. Show that the set of all finite sets of integers is a countable set.

Answer: The set of all finite sets of natural number is given here. All integers is similar.
Here is an enumeration of these sets in steps 1,2,3,..., k, .... below

Step 1. We enumerate first all subsets \{1\}. There are 2 of these. ...

Step k: Enumerate all subsets of \{1, 2, 3, ..., k\}. There are \(2^k\) of these. ....
At each step only finitely many sets are enumerated. Every finite set of integers is enumerated in some step, actually in infinitely many steps, and so every finite set of integers gets enumerated, and the whole enumeration is countable.

6. Show that the recognizable languages are closed under concatenation. That is, if J and L are two languages that are recognizable, then so it the language JL. (JL is the concatenation of J and L.)

Answer: This one is a bit tricky. It is easier though if you use the equivalence between a language being recognizable and enumerable.

As J and L are recognizable, they are enumerable. Let \(j_1, j_2, ..., l_1, l_2, ...\), be enumerations of J and L. We show how to enumerate JL.

Step k of the enumeration is:
List all pairs of string JL, where \(j\) is a string in \(j_1, j_2, ..., j_k\) and \(l\) is a string in \(l_1, l_2, ..., l_k\). (There are \(k^2\) such strings.)

As you do this only strings in JL are enumerated and every string in JL is enumerated at some point in the list. So JL is enumerable and hence recognizable.

7. Consider the following nondeterministic Turing machine N where
N has states \(q_i\) for \(i = 1, 2, 3, 4, a, b\). N’s input alphabet is \(\{0, 1\}\), N’s tape alphabet is \(\{0, 1, B\}\),

N’s has the following \(\delta\) function as it’s program.
Note: If the definition of \(\delta\) is not specified then you may assume the program simply halts and is in the \(q_r\) state when it does. So for example if at some point the TM is in state state \(q_2\) and reads a 0 then it goes to state \(q_r\) and halts.

So here is \(\delta\):
\[
\delta(q_0, 0) = \{(q_0, 0, R)\}
\delta(q_0, 1) = \{(q_0, 1, R), (q_1, 1, R)\}
\delta(q_1, 0) = \{(q_0, 0, R)\}
\delta(q_1, 1) = \{(q_2, 1, R), (q_3, 1, R)\}
\delta(q_2, 1) = \{(q_4, 1, R)\}
\delta(q_3, 0) = \{(q_4, 0, R)\}
\]
\( \delta(q_1, 1) = \{(q_a, 1, R)\} \)

i. Run the TM on the input strings 001101 and on 1001. Does \( N \) accept either of these strings? Explain why.

ii. Describe which strings are in the language \( L(N) \), where \( L(N) = \{ w \mid \text{the machine } N \text{ accepts } w \} \)?

Answer: None given here. This example will be gone over in class quite soon.