1. Prove the correctness of RadixSort by (finite) induction on $i$, the column being sorted. Be sure to carefully state your induction hypothesis. It should be clear from your proof why the sort on line 2 must be stable, and why the algorithm must sort the digits from least to most significant.

**RadixSort**$(A, d)$ (Pre: $A[1..n]$ consists of $d$ digit numbers)
1. for $i \leftarrow 1$ to $d$
2. sort $A$ on digit $i$ using a stable sort

2. Prove that Counting Sort is stable. Also show that if we reverse the order in which the final loop is executed, the resulting algorithm is correct, but not stable. (This is problem 8.2-3 on page 170).

3. Recall the coin changing algorithm described in class:

**CoinChange**$(d, N)$ (Pre: The supply of coins in each denomination is unlimited)
1. $n \leftarrow \text{length}[d]$
2. for $i \leftarrow 1$ to $n$
3. $C[i, 0] \leftarrow 0$
4. for $i \leftarrow 1$ to $n$
5. for $j \leftarrow 1$ to $N$
6. if $i = 1$ and $j < d[i]$
   7. $C[1, j] \leftarrow \infty$
8. else if $i = 1$
9. $C[1, j] \leftarrow 1 + C[1, j - d[i]]$
10. else if $j < d[i]$
11. $C[i, j] \leftarrow C[i - 1, j]$
12. else
13. $C[i, j] \leftarrow \min(C[i - 1, j], 1 + C[i, j - d[i]])$
14. return $C[n, N]$

a. Write a recursive algorithm which given the table $C[1...n; 0...N]$ created by the above algorithm, prints out a sequence of $C[n, N]$ coin values which disburse $N$ monetary units. In the case $C[n, N] = \infty$, print a message to the effect that no such disbursement is possible.

b. Modify the above algorithm to allow for the possibility that the supply of coins in some denominations is limited. Let $L[1..n]$ be an input array giving the limits on each denomination, i.e. at most $L[i]$ coins of denomination $d[i]$ are to be used ($1 \leq i \leq n$), where $0 \leq L[i] \leq \infty$. 