Question 1 (Knapsack, 30 points)**.** *Consider the knapsack problem where you are allowed to take items more than once, have a total capacity of 10 and have the following items available:*

Item	*Weight*	*Value*
A	*2*	*4*
B	*3*	*5*
C	*4*	*9*
D	*5*	*10*
E	*7*	*16*

What is the best combination of items to take?

Running the dynamic program covered in class we get the following table:

Capacity	0	1	2	3	4	5	6	7	8	9	10
Value	0	0	4	5	**9**	10	13	16	**18**	20	**22**

Working backwards, we find that 22 can be achieved either as a copy of A and items of weight 8 or item C plus items of weight 6. We consider the former case. The 18 for items of weight 8 can only be achieved by a copy of item C plus items of weight 4, and the optimum there can only be achieved by a single copy of C. Thus, the best combination of items is one copy of A and two copies of C.

Question 2 (Business Plan I, 35 points). *Harold is starting a new business. He begins with* \$0 *in capital and wants it to be worth as much as possible at the end of n weeks, at which point he plans to sell. In order to get there, he has k different strategies he can use to make money. Each strategy takes one week to implement, and can be used multiple times as needed. If Harold has x dollars at the start of a week and implements the i^{th} strategy that week, he will have $f_i(x)$ dollars at the end of that week for some increasing function f_i (meaning that $f_i(x) \geq f_i(y)$ whenever $x \geq y$).*

Harold has a simple plan for determining which strategies to use in which order. Each week he will note the number of dollars, x, that he has at the start of the week, and implement the plan with $f_i(x)$ as large as possible. Prove that this strategy gets Harold as much money as possible by the end of the n^{th} week.

We prove by induction on t that this strategy has at least as much money at the end of week t as any other strategy. For a base case we use $t = 0$ and note that all strategies have 0 dollars at the start. Assuming that we know that our strategy has the greatest possible amount of money, x, at the end of week t, we will show that it has the greatest amount of money at the end of week $t+1$. Suppose that Harold's strategy uses the i^{th} strategy on week $t+1$ to end with $f_i(x)$ dollars. Suppose that another schedule ends with y dollars at the end of week t and uses the j^{th} strategy during week $t+1$ to end with $f_j(y)$ dollars. We have that $f_i(x) \geq f_j(x)$ by the method that Harold used to select i. We have that $x \geq y$ by the inductive hypothesis. We therefore, have that $f_j(x) \geq f_j(y)$ since f is increasing. Therefore $f_i(x) \geq f_j(x) \geq f_j(y)$, proving our inductive hypothesis.

Question 3 (Business Plan II, 35 points). *The setup of this problem is similar to that in question 2. Please read that question first.*

Harold later realizes that things are not quite as predictable as he had previously thought. He determines that implementing plan i in a given week has a probability $p_{i,j}$ of earning him j dollars (adding j to his current amount of money) over the course of that week for each integer j between 0 and m. Harold starts with 0 dollars and his goal is to maximize the probability that he will have at least m dollars at the end of n weeks.

Give an algorithm that given n, m and all of the $p_{i,j}$ computes the maximum possible probability with which Harold can achieve this goal. For full credit, your algorithm should run in time $O(nm^2k)$ where k is the number of possible strategies available.

We let $B(d,t)$ be the best probability that Harold can achieve if he has d dollars at the end of week t. We note that $B(d,n)$ is 0 if $d < m$ and 1 otherwise (since at this point, he either has enough money or doesn't). We also know that $B(d,t) = 1$ if $d \geq m$. Otherwise, if Harold has d dollars at the end of week t and uses the i^{th} strategy on week $t+1$ he will have $d+j$ dollars with probability $p_{i,j}$. If he uses the optimal strategy from then onwards, his probability of success will be $B(d+j,t+1)$. Thus, Harold's overall probability of success will be $\sum_{j=1}^{m} p_{i,j} B(d+j,t+1)$.

This means that in order to optimize his probability of success, Harold should use the strategy i that maximizes $\sum_{j=1}^{m} p_{i,j} B(d+j,t+1)$. Thus, $B(d,t) = \max_i(\sum_{j=1}^{m} p_{i,j} B(d+j,t+1))$. This gives rise to the natural dynamic program

```
BestOutcome(n,m,p)
  Create Array B[0..2m,0..n]
  For t = n to 0
    For d = 0 to 2m
      If d >= m
        B[d,t] = 1
      Else if t = n
        B[d,t] = 0
      Else
        B[d,t] = Max{sum_j p_{ij}B[d+j,t+1]}
  Return B[0,0]
```

To show correctness, we prove that each time $B[d,t]$ is assigned a value, it is assigned the correct value of $B(d,t)$. This can be proved by induction using the base case and recursion proved above. We note that $d+j$ is always at most $2m$ and since we assign values for larger values of t first, $B[d+j,t+1]$ will always be assigned before $B[d,t]$ is. Thus, $B[0,0]$ will eventually be assigned the correct probability of success if Harold has no money at the start of week 0.

To analyze runtime, we note that the main loop has $O(nm)$ iterations. Each iteration takes constant time except for the computation of the maximum. Each term in the max is a sum over $O(m)$ things and we need to consider $O(k)$ many possible values of i. Therefore, this step can be done in $O(mk)$ time. Thus, the total runtime is $O(nm^2k)$.