1. Your factory uses a supply of parts at a steady rate of 10 per month. You have inventory capacity to hold 40 of the parts, and holding parts in inventory for one month incurs a direct cost of $0.50 per part per month. Your firm also has an internal cost of funds of 0.4% per month. At the beginning of each month, you may place an order for parts, which cost $200 each, plus an overhead cost of $50 for each order placed.

You have very stringent quality requirements for the parts, which your supplier, despite being the best one available, often does not meet. Each part ordered has an independent 7% chance of being not meeting your standards. You immediately return such rejected parts for a full refund, but shipping them back to the supplier costs $5 per part.

Along with parts already in inventory, parts ordered at the beginning of the month may be used to meet the current month’s usage, so long as they are not rejected. If you fall short of the usage requirement for a given month, each part you are “short” that month incurs a $450 opportunity cost.

With a Python program, use dynamic programming to compute the part ordering policy that minimizes the expected present value of your costs over the next 12 months, under the constraint that you are not permitted to overflow your inventory capacity. Assume that parts left in inventory at the end of the 12 months have a salvage value of $180 each, and you start with an inventory of 13 parts.

Hand in a printout of your Python program and a printout of its output. Also submit your Python code to Sakai under the “Assignments 2” tab, “Assignment 6 (only one problem)”. 