Due March 21, 2012

Goals:

Understanding of the five steps for developing a dynamic programming solution.

Requirements:

1. Write a C program to extend the dynamic programming solution to the parking permit problem (Notes 07.H.) with a secondary market of permits with fixed beginning and ending days, along with a cost. The goal is still the same - to produce a minimum cost solution using the (original) flexible permits with unspecified beginning and ending days, along with the inflexible secondary market permits.

   The input should be read from standard input:

   1. The first line will have the values of k, n, and m. k is the number of (flexible) permit types. n is the number of days for which parking is needed. m is the number of secondary market permits. Either k or m may be zero.
   2. The next k lines will each have a pair of values for a permit type: the number of days covered and the cost. These pairs will appear in strictly increasing order for the number of days and likewise for the costs.
   3. The next n lines will each contain an integer corresponding to a day you must park. These values appear in strictly increasing order.
   4. The last m lines will each contain a triple for a secondary market permit: beginning day, ending day, and cost. The beginning day will not be after the ending day. These permits are not required to be ordered.

   The output must include 1) the table of subproblems, along with backtrace information, 2) the cost of the final solution, and 3) the sequence of permits needed and the range of days covered.

2. Email your program to `yuanzhe.cai@mavs.uta.edu` by 3:45 p.m. on March 21, 2012. The `Subject` should be your name as recorded by the University and you should `cc:` yourself to verify that you sent the message correctly.

Getting Started:

1. You may extend the program `parkingOff.oneBased.c` available on the course webpage.

2. The most significant extension to the code is for checking secondary market permits for each of the parking days. A simple solution would check all of these permits to see if the day for the current subproblem is within range. A better solution (that will avoid a point penalty) uses sorting (use any sort you wish) and additional tables to limit the secondary market permits examined for each parking day. Note, however, that like the original problem, all k of the flexible permit types must be examined for each day.

3. The secondary market permits raises the possibility of overlapping permits in the solution.