CSCI-650 **Project 2**

You will implement the dynamic programming algorithm for solving 0-1 Knapsack problem.

Requirements:

I) Your algorithm will read from a user three lines (use *cin*):

1) On the first line there will be two integers. For example:

5 20

The first integer is the number of items in the store. The second integer is the capacity of the available knapsack.

2) Next line has values for each of *n* items.

3) The third line has weights for each of *n* items.

```
3 7

2 6 4

1 5 3
```

In the given above example of a user's input, there are 3 items, knapsack's capacity is 7. The value of the first item is 2 and its weight is 1. The value and weight of the second item are 6 and 5 respectively. The value and weight of the third item are 4 and 3 respectively.

II) Then your program will run dynamic programming algorithm to fill two 2-dimensional arrays (or vectors): one for best values and the other for backtracking.

III) Then your program will run backtracking to retrieve the selected items. Use ***recursive function*** for backtracking.

IV) Output on the first line the total value of the selected items in the knapsack and on the second line print out the selected items in increasing order (count of items starts with 1) separated by a space and use *endl* at the end of each lines.

For the given example, your output will be:

```
8
1 2
```

Submission: via turnin system.

Deadline: Thursday, March 24, by 11:59PM.