User documentation for project 2
Usage: proj2 [-s][-v][-h][-f <filename>]
-s: silent mode
-v: verbose mode
-h: print this help
-f <filename>: read <filename> into the current set

Menu options:

0. exit
1. input file <filename>: open and read a list from a file to the current multiset
2. union file <filename>: open and union a multiset from a file with the current multiset
3. subtract file <filename>: open and subtract multiset from a file from the current multiset
4. difference file <filename>: open and find the difference between a multiset from a file and the current multiset
5. intersect file <filename>: open and find the intersection between a multiset from a file and the current multiset
6. reset current multiset to the empty multiset
7. output file <filename>: open and write the current multiset to a file
8. print current multiset to the console
9. find <item name>: test if <item name> is in the current multiset
10. insert <item name> <count>: add <item name> to the current multiset with number <count> if <item name> is not in the current multiset, or increase <item name>'s number by <count> if it is
11. delete <item name>: remove <item name> from the current multiset if it is in it
12. reduce <item name> <count>: reduce the number of <item name> by <count>
13. verbose output
14. normal output
15. silent output
16. help
17. max <filename>: open and find the maximum between a multiset from a file and the current multiset.

Synopsis:

1. The input operation will empty the current multiset and fill it with the multiset read from the specified file, and write “New multiset loaded” to the console.

2. Union will read in the multiset from the file; if item “X” with number n is read from the file, and item “X” is not in the current multiset, then “X” will be added to the current multiset with the count n; otherwise if item “X” has current count m, its count is updated to m+n. When the union operation is done, the system will print “<file> union completed” to the console.

3. Subtract will read in the multiset from the file; if item “X” with number n is read from the file, and item “X” with count m > n is in the current multiset, then “X” will be remain with count m-n; if n >= m, then “X” is removed from the current multiset; otherwise nothing happens.

When the subtract operation is done, the program will print “<file> subtraction completed” to the console.

4. Difference will read in the multiset from the file (note that multiple appearances of the same item must be handled upon reading the file in). If item “X” with number n is read from the file, and item “X” is not in the current multiset, then “X” will be added to the new current multiset with number n; likewise, if “Y” is in the current multiset with number m and “Y” is not in the multiset read from the file, then “Y” with number m will be added to the new current multiset. If “X” with number n is read in, and “X” with number m is in the current multiset, then the new current multiset will have “X” with the number |m – n| unless m = n, in which case “X” is not in the new current multiset. When it is done, the program will print “<file> difference completed” to the console.

5. Intersect will read in the multiset from the file; if item “X” with number n is read from the file, and item “X” with number m is in the current multiset, then “X” with the number min(m, n) will be added to the current multiset, otherwise “X” is not in the new current multiset. When it is done, the program will print “<file> intersection completed” to the console.

6. Reset will remove all items from the current multiset and will print “Reset completed” to the console.

7. Output will save the current multiset into a file in the same format as a multiset read from a file. When it is done, it will print “Save to <file> completed” to the console.

8. Print will perform this operation to the console, but will print “Current multiset:” followed by a newline, then the items, rather than the completion statement.

9. Find <item> will search the current multiset and print “Item <item> found with count <count>” if <item> is in the current multiset with number <count>; otherwise it will print “Item <item> not found”.

10. Insert <item> <count> will add <item> to the current multiset with number <count> if it is not already in the current multiset, and will change the number to m + <count> if <item> is already in the multiset with number m. The program will print “Item <item> inserted with count <m'>” where m' is the final value of the number.

11. Delete <item> will remove <item> from the current multiset if it is present, and will print “Item <item> deleted”, or will print “Item <item> not in multiset” otherwise.

12. Reduce <item> <count> will reduce the number m associated with <item> by <count>, or will remove <item> if the resulting number m - <count> is less than or equal to zero. The program will print “Item <item> count reduced to <m – count>” if the new number is greater than zero, or will print “Item <item> removed” if the new number is less than or equal to zero, or will print “Item <item> not in multiset” otherwise. Note that reduce with a negative number is allowed, and results in increasing the number associated with the item.

13. Verbose will cause the current state of output prompt to remain verbose until changed. The default state is normal. In verbose state, the user prompt shall include a short list of all

commands (i.e., the numbers and a one-word description along with parameter if any). In verbose state, additional information as you see appropriate (and at least some additional information for each of the commands that change the state of the current multiset) shall be printed to the console.

14. Normal will cause the current verbosity state to remain normal until changed. In normal state, the user prompt shall be “> “ on a new line without a newline at the end. This prompt will also be printed in verbose state. In both cases, the feedback indicated for each command above will be printed to the console.

15. Silent will cause the current verbosity state to remain in silent mode until changed. In silent mode, there is no prompt and there is no output to the console. Remember that error messages are printed to standard error.

16. Help prints a more complete description of the commands and their meanings, followed by the regular “> “ prompt (once), regardless of the current verbosity mode.

17. Max is similar to union, but with an important difference: the numbers are not added, but rather the larger of the two is used as the result. Max will read in the multiset from the file; if item “X” with number n is read from the file, and item “X” is not in the current multiset, then “X” will be added to the current multiset with the count n; otherwise if item “X” has current count m, its count is updated to the maximum of m and n. When the max operation is done, the system will print “Max completed” to the console.