CMSC 401 Spring 2022- Programming Assignment 1 (HW1): (corrected 2/22, see green below)

Max points: 10

Problem Description:

Write a program that implements a **min-heap** (a heap where lowest priority key has highest priority). Your implementation should handle three operations:

Insert
DecreaseKey (the min-heap equivalent of IncreaseKey)
ExtractMin (the min-heap equivalent of ExtractMax)

The heap will need to contain simple two-element objects: each object will have a unique ID (an integer from 0 to some N-1), and a (possibly non-unique) priority (an integer between 1 and 10000).

At the start of the program, the heap is empty. The program should manage the heap based on inputs from System.in. The input will be composed of lines.

First line will contain one integer, N (1<=N<=10000). All object IDs are guaranteed to be lower than N (that also means there will be at most N objects in the heap).

Each subsequent line is a command. There are four possible commands:

- Insert. Format: "`1 x y`". The program should create object with ID=x and priorityKey=y, and insert it into the heap. Nothing is being printed to the screen. For each unique ID x, there will be at most one Insert command.
- DecreaseKey. Format: "`2 x z`". The program should decrease the priority key of object with ID equal to x, from its current priority to new priority equal to z, and update the heap to reflect this. Nothing is being printed to the screen.
- ExtractMin. Format: "`3`". The program should extract the minimum from the heap, and print the extracted object's ID and current priority to the screen, as two integers separated by space.
- Exit. Format: "`4`". Exit the program.

Examples (the empty lines on output below are just to make understanding easier, your program should not print them):

Input	Output	Comments for easier understanding (your program should not print them)
5		# there will be at most five elements (IDs 0,…,4)
1 4 40		# insert ID=4, priority 40
3	4 40	# extract min: ID=4, priority=40 extracted
1 2 14		# insert ID=2, priority 14
2 2 8		# change priority of ID=2 from 14 to 8
3	2 8	# extract min: ID=2, priority=8 extracted
4		# exit
Input	Output	Comments for easier understanding (your program should not print them)
4		# there will be at most four elements (IDs 0,…,3)
1 2 20		# insert ID=2, priority 20
1 1 11		# insert ID=1, priority 11
1 3 28		# insert ID=3, priority 28
1 0 44		# insert ID=0, priority 44
3	1 11	# extract min: ID=1, priority=11 extracted
3	2 20	# extract min: ID=2, priority=20 extracted
2 0 6		# change priority of ID=0 from 44 to 6
3	4 6	# extract min: ID=0, priority=6 extracted
3	3 28	# extract min: ID=3, priority=28 extracted
4		# exit

Submissions:
Due date: **Wednesday, March 2nd, 5pm**. Submit through Canvas: **upload a single java source code file, named cmsc401.java (your class should be named cmsc401)**. Include your name in a comment at the top of the file. Do not use packages in your code. The command *javac cmsc401.java* should be sufficient to compile your code, and *java cmsc401* to run it.

Note:

- You should assume that inputs are always correct. No “wrong intput” handling needed.
- If you want, you can use the skeleton code (cmsc401.java) provided to you for taking input and writing output.
- You should assume there will be one test case at a time. You do not need to think about handling multiple test cases. Grading will be done using automated script. It will be handled there.
- Do not print extra lines such as: “Please enter..”, “The output is..”, etc.