CSC 225 SUMMER 2016
ALGORITHMS AND DATA STRUCTURES I
ASSIGNMENT 3 - PROGRAMMING
UNIVERSITY OF VICTORIA

1 Programming Assignment

Write a program that prints out all integers of the form $a^3 + b^3$, where a and b are integers in the range $[0, n]$, in sorted order while using $O(n)$ space. That is, you cannot use an array of size n^2 and then sort it.

> **Input:** A nonnegative integer *n*.
> **Output:** A sorted list of all integers of the form $a^3 + b^3$, where a and b are integers in the range $[0, n]$.

For each value integer of this form you will need to keep track of three things, the integer itself, $a^3 + b^3$, and the two integers used to create it, a and b. So, let's denote our elements with the triples $(a^3 + b^3, a, b)$. We do not need duplicate values ($a^3 + b^3 = b^3 + a^3$) so we can assume that $a \geq b$.
Hint: You can start with the $n + 1$ values, $(0^3 + 0^3, 0,0), (1^3 + 0^3, 1,0), \dots, (n^3 + 0^3, n, 0)$. At any given time if the current smallest element is $(i^3 + j^3, i, j)$ and $j < i$, then you may print $i^3 + j^3$, remove it from your data structure and add new element $(i^3 + (j + 1)^3, i, j + 1)$. Otherwise, you may just print it, remove it and then find the next smallest element.

Your task is to write a java program, stored in a file named SortedSumsOfCubics.java that contains a function SortedSumsOfCubics, which takes a nonnegative integer *n* as its only argument, and prints out all the integers of the form $a^3 + b^3$, where a and b are integers in the range $[0, n]$, in sorted order. For example, if the input is $n = 3$, the output should be,

0 1 2 8 9 16 27 28 35 54

The main function in your code should help you test your implementation by getting an integer from input and sending it to the function above.

2 Evaluation Criteria

The programming assignment will be marked out of 25, based on a combination of automated testing (various values for *n*, some very large) and human inspection.

Score (/50)	Description
0 - 5	Submission does not compile or does not conform to the provided description.
5 - 15	The implemented algorithm uses $O(n^2)$ space or is inaccurate. That is, the outputs are not in sorted order or some are missed.
15 - 20	The implemented algorithm uses $O(n)$ space and $O(n^3)$ or worse running time or minor errors.
20 - 25	The implemented algorithm uses $O(n)$ space and $O(n^2 \log n)$ running time and is accurate.

To be properly tested, every submission must compile correctly as submitted. **If your submission does not compile for any reason (even trivial mistakes like typos), or was not based on the template, it will receive at most 5 out of 25.** The best way to make sure your submission is correct is to download it from conneX after submitting and test it. You are not permitted to revise your submission after the due date, and late submissions will not be accepted, so you should ensure that you have submitted the correct version of your code before the due date. conneX will allow you to change your submission before the due date if you notice a mistake. After submitting your assignment, conneX will automatically send you a confirmation email. If you do not receive such an email, your submission was not received. If you have problems with the submission process, send an email to the instructor **before** the due date.