CSC 225 SUMMER 2014
ALGORITHMS AND DATA STRUCTURES I
ASSIGNMENT 1 - PROGRAMMING
UNIVERSITY OF VICTORIA

1 Programming Assignment

The assignment is to design and implement an algorithm for the TripleSum225 problem. The problem is defined as follows:

Input: An array A of n non-negative integers.

Output: A boolean value (`true` or `false`). If there are three indices i, j, k such that $0 \leq i, j, k \leq n-1$ and $A[i] + A[j] + A[k] = 225$, the output will be `true`. If no such indices exist, the output will be `false`. Note that i, j and k do not have to be distinct.

A Java template has been provided containing an empty function `TripleSum225`, which takes an integer array A as its only argument, and returns a boolean value. Your task is to write the body of the `TripleSum225` function. You may assume that the provided array A conforms to the specification above (that is, A will contain only non-negative integers).

You must use the provided Java template as the basis of your submission, and put your implmentation inside the `TripleSum225` function in the template. You may not change the name, return type or parameters of the `TripleSum225` function. The `main` function in the template contains code to help you test your implementation by entering test data or reading it from a file. You may modify the `main` function, but only the contents of the `TripleSum225` function will be marked. Please read through the comments in the template file before starting.

2 Examples

The table below shows the correct output of the `TripleSum225` function on various test inputs. In cases where the output is `true`, a set of three elements of the array which sum to 225 is shown. Note that there may be more than one triple of elements which sum to 225.

Input Array	Result	Triple
50, 100, 75, 500	`true`	$50 + 75 + 100 = 225$
150, 125, 100, 175	`false`	(none)
1, 200, 100, 225	`false`	(none)
50, 25, 75	`true`	$75 + 75 + 75 = 225$
225, 500, 1000	`false`	(none)
225, 0, 1, 5000	`true`	$0 + 0 + 225 = 225$
224, 1, 2, 2, 2	`false`	(none)
224, 1, 0, 2, 2	`true`	$0 + 1 + 224 = 225$

3 Test Datasets

A set of input files containing test data are available under the 'Data' tab on conneX, sorted by their size and whether or not they contain a triple which adds to 225. You should ensure that

your implementation gives the correct answer on these test files before submitting. It may also be helpful to test your implementation on a variety of other inputs, since the posted data may not cover all possible cases. Depending on the running time of your algorithm, it may not be able to process some of the larger input files.

4 Evaluation Criteria

The programming assignment will be marked out of 50, based on a combination of automated testing (using large test arrays similar to the ones posted on conneX) and human inspection.

There are several possible algorithms for the TripleSum225 problem, with a variety of running times. For an input array containing n values, the simplest algorithm is $\Theta(n^3)$ and the optimal algorithm is $\Theta(n)$. The mark for each submission will be based on both the asymptotic worst case running time and the ability of the algorithm to handle inputs of different sizes. The table below shows the expectations associated with different scores.

Score (/50)	**Description**
$0-5$	Submission does not compile or does not conform to the provided template.
$5-25$	The implemented algorithm is $O(n^3)$ or is substantially inaccurate on the tested inputs.
$26-40$	The implemented algorithm is $O(n^2 \log n)$. Input arrays of size 10000 can be processed in under 30 seconds, but arrays of larger sizes may not be feasible in a reasonable amount of time.
$41-50$	The implemented algorithm is $O(n)$, gives the correct answer on all tested inputs, and can process arrays of size 1000000 in under 10 seconds.

To be properly tested, every submission must compile correctly as submitted, and must be based on the provided template. **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 50**. 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.