COMP 482 Project 1
Due: Friday October 9 2355

Task: You will write a java program which

- reads a set of integer data from a file,
- stores that data into a data structure, and
- **repeatedly** determines whether any 3 items from the data set sum to a target value.

Your program must:

- compile using the command 'javac Project1.java',
- run using the command 'java Project1',
- accept input from a file called input1.txt formatted precisely as described below,
- output results exactly as described below, and
- run in time $O(n^2)$.

If your submission fails to do any of these things, expect a score no higher than 5/20.

Your program should:

- be properly formatted (eg using proper indentation)
- be structured reasonably (eg dividing the tasks up into methods), and
- be commented appropriately (eg containg your name at or near the beginning of each file and explaining how/why a method works just above the method).

If your submission fails to do any of these things, expect a score no higher than 15/20.

Input Format: After downloading your submission, I will place a file called input1.txt into the same directory. The file will begin with strictly positive integers seperated by whitespace (spaces/tabs/newlines). These are the values you should place into a data structure. The file will then have a zero value (ie 0) which indicates the values which should be placed in the data structure have finished. The file will then have integer target values seperated by white space. For each of these final integers, your program should determine whether any 3 of the items in the data structure sum to the target value.

Sample input file - input1.txt:

```
15 82 22 36 29
11 31 4 31
0
119 9 -2 57
29 73 93
```

Output Format: For each of the target values you should output to the screen 1 line consisting of the target, a single space, and either YES or NO (depending on whether the target is the sum of three values). Explanation of the output below: some target values are the sum of 3 of the initial values 119=15+82+22, 57=15+11+31, 73=11+31+31 (31 can be used twice because it occurs twice, but cannot be used 3 times because it does not appear 3 times), but others are not the sum of three values.

Sample output:

```
119 YES
9 NO
-2 NO
57 YES
29 NO
73 YES
93 NO
```

Stray Thoughts:

I will be using a recent version of Java (likely the current version Java SE 15, but if Oracle releases a new version I may upgrade).

You'll be submitting only dot-java files (no class files or input files required or wanted).

You are allowed to use any of the standard features, classes, methods in Java. For example, I expect nearly all students will want to use either an array or java.util.ArrayList and the built in sort routine (either for arrays or ArrayLists). This is allowed.

You can use as many or as few files as you feel appropriate, but the main method should be located in a file called Project1.java. Otherwise the project won't compile/run with the required commands.

Some IDEs default to placing java files into packages. This will likely cause the commands 'javac Project1.java' and/or 'java Project1' to fail. Either use an IDE that does not place java files into packages OR learn your preferred IDE well enough to avoid this issue OR delete any package lines before submission.

Students often decide to change or modify the format of the input or ouput. Sometimes it makes the project easier for them. Other times a student thinks it is an improved design. You may or may not be right, but don't change the input or output format. Doing so will result in your project getting a low score.

It is likely that many students won't read this far. There is no need to let me know you've read this.

There is a very easy way to determine answers in time $O(n^3)$. You should not use this method. You should instead use a method which is $O(n^2)$.

I will likely use the sample input file above while grading, but I'm also likely to use other much larger input files (possibly containing millions of data items, but no more than the JVM limit on arrays $\approx 2^{30}$).

I suggest you finish you project several days in advance. This way you have time and opportunity to ask any last questions and verify that what you upload satisfies the requirements.

Your project should be written and understood by you. Helping or receiving help from others is allowed, but significant shared source code indicates that you either did not write/understand what you submitted or you assisted another in submitting code they did not write or understand.

These project description files are getting excessively long, but I find that failing to state things which are common sense yields complaints.