## Objectives:

1. Learning how to read and write files, interpreting the input as bytes, characters, or int
2. Recalling concepts from prog 1, namely *casting* and *try-catch*
3. Dealing with multiple files at once
4. Decoding a super secret hidden message

## Submission:

* Make sure that the name of all team members is included in all files you will submit
* One member of your team should email me directly with your code. CC all your teammates. Feedback will come :)

## Task 1 - practice round:

In the ALCFileUtil, implement a **static** method called readCharacterN. This method should take as parameters:

* A String representing the name of a file
* An int, representing which character from the file we want.

Imagine our file **Test.txt** contains the following:

Abcd

Efgh

Ijklmn

Calling readCharacterN(“Test.txt”, 0) should return ‘A’.

Calling readCharacterN(“Test.txt”, 2) should return ‘c’.

Calling readCharacterN(“Test.txt”, 4) should return a new line character.

Calling readCharacterN(“Test.txt”, 5) should return ‘E’.

### Resources & hints:

1. Javadocs: <https://docs.oracle.com/javase/8/docs/api/java/io/InputStreamReader.html#read-->
2. Note that there are 2 read methods, you can solve this using either of them. Think about how each one works, and synchronize with your team.

## Task 2 - the secret files:

Our message will be hidden in files that follow this pattern:

{int}

{bunch of text, with newlines etc.}

In the ALCFileUtil, implement a **static** method called findSecretCharacter. This method should take as parameters a String representing the name of a file. You may assume that the file will be in the format above.

Your first task will be to read the int in the first line. Let’s call it **n**

You should then return the **n**th character from the text portion of the file

Here are some examples. If the file looks like

3

Abcd

Efgh

Ijklmn

We should return the character ‘d’

If the file looks like

0

Abcd

Efgh

Ijklmn

We should return the character ‘A’

### Resources & hints:

1. <https://docs.oracle.com/javase/7/docs/api/java/io/DataInputStream.html>
2. The class above lets you read both int and char values!
3. It’s actually a bit hard to use your method from task 1 directly. I recommend you copy-paste some of that logic for now, we can discuss how to design this differently in the solution

## Task 3 - finding the secret message:

Let’s bring it all together!

In the ALCFileUtil, implement a **static** method called findSecretMessage. This method will take no parameters.

On the day of the dojo, you will be provided with 14 different files. The secret message is scattered across them. Luckily, the files are named in a helpful way: “secret1”, “secret2”, “secret3” …. All the way up to “secret14”. This should make it easy to loop over all the files.

Each file is in the format described in Task 2, and will contribute a single character to the secret message.

Your method should grab the hidden character from each of these files in order, and write it to a file named “SecretMessage.txt” If the message makes sense, you have probably done this right :)

### Resources & hints:

1. The hard part of this task is already behind us, in Task 2. Take a deep breath, think about this, you and your team got this.
2. <https://docs.oracle.com/javase/8/docs/api/java/io/OutputStreamWriter.html>