Programming in Java
MTech Preparatory Term - 2019
Lab 1

Garbage Collection

Run the following piece of code:

```
public class Test1 {
        public static void main () {
                int N = 10;
                int M = 100000;
                for(int i =0; i< N; i++) {
                        int[] box = new int[M];
                }
        }
}
```

Are there large values of N (within the range of “int”) for which this program does not work?

If you modify the program as follows:

```
public class Test1 {
        public static void main () {

                int N = 10;
                int M = 100000;
                int[][] boxes = new int[N][];
                for(int i =0; i< N; i++) {
                        int[] box = new int[M];
                        boxes[i] = box;
                }
        }
}
```

How large can you make N and still be able to run on your machine? Why is this different from the first version?

Can you re-configure Java on your machine to allow larger amounts of memory (heap)?

Matching Parenthesis

Write a Java program to read in a file containing a sequence of parentheses of various kinds - (){}[] - in any order and combination, and check if the parentheses are “balanced”.
In the case of failure, the program should specify the position in the input stream where the first mismatch is detected (the first input character is at location 0)

For example, for the input,
([]{()})
the output should be **success**
and for the input
([]{())}
the output should be **failed at 6**

Implement this without using explicit arrays/lists/stacks or other heap memory.

Image Labeller: Mini project (to be completed over the duration of this course)

We would like to develop a simple interactive application that allows a user to browse through the photos in a local directory and create an album of selected files, with a caption for each selected file. We will assume that the photos cannot be modified, hence, the captions will be written to a file, along with other information relevant to the album.

Broadly, the user can do the following:

1. Start a new album in a given directory
2. View the list of files that are photos/images. Sort by name or date of file or other properties.
3. View a selected image. Decide if it should be added to the album (or removed)
4. Add/modify the caption
5. Go to the next or previous image
6. … (to be added)

We would like to design this with a clear separation of the data (or **model**) from the UI (or **view**). As a first part, design the model - Album, Photo etc, how they will be organized, the interfaces that would be required, etc. We will add or modify some of the features listed above as we go forward. (You are welcome to suggest new requirements!)

Design a simple command-line interface that will allow you to test the model (apart, of course, from viewing the actual image).
In the next week, you will be provided with a simple (and crude) Java-based UI (implemented in Swing) that you can integrate with the model, and can extend to enhance functionality.

Optionally, you can look at converting this to a REST server, and build a browser-based UI (using your favourite framework!)