

## Trial Lab Test 1

**[20 marks]****General Instructions:**

1. **You are not allowed to communicate in any way during the test. Any violation of this instruction will result in a zero score for your lab test.**
2. You will perform the lab test on your personal laptop.
3. You can refer to any file on your laptop.  
**Failure to do the following will attract a penalty up to 20% of your score for that question.**
4. Make sure your code can generate exactly the same output as we show in the sample runs. You may get some marks deducted for missing spaces, missing punctuation marks, misspelling, etc. in the output.
5. Do not hardcode. We will use different test cases to test and grade your solutions.
6. Follow standard Java coding conventions (e.g. naming of getter and setter methods, choice of identifier names for classes, methods and variables) as well as indent your code correctly.
7. Tools that enable AI pair programming (e.g., GitHub co-pilot) or pair programming (Live Share) are not allowed.
8. Ensure that all your Java code can compile without compilation errors. They must compile with any test class(es) that are provided. You may wish to comment out the parts in your code, which cause compilation errors. But remember that commented code will NOT be graded. **Only code without compilation errors SHALL be graded.** You may need to comment out parts of the test class' code that you have not implemented in order to test your implemented solutions.
9. Download the source code and rename the root folder in the zip file to your **Email ID (e.g. sotong.tan.2022)**
10. Include your name as author in the comments of all your submitted source files. For example, if your registered name is "Sotong TAN", include the following block of comments at the beginning of each source file (.java) you write.

```
/*  
 * Name: Sotong Tan  
 */
```

**Question 1****[ 8 marks ]**

Find the file **Q1.java** in the folder <your email ID>\labtest1\Q1\ . Insert your name and email ID to the top of Q1.java.

Implement the **getSmallestPair** method in **Q1.java**. When the **getSmallestPair** method is invoked with **num**, it returns the integer which is the SMALLEST of all possible pairs of 2 consecutive digits formed from **num**.

**Note:** You can assume **num** is a positive number.

For example,

1. if n is 2345, the possible pair of digits are: 23, 34, 45. The smallest pair is 23
2. if n is 10245, the possible pair of digits are 10, 2 (formed because of 02), 24, and 45. The smallest pair is 2.

number	Possible Pairs	Positions in number
2345	23	<u>2</u> 345
	34	2 <u>3</u> 45
	45	234 <u>5</u>
10245	10	<u>1</u> 0245
	2	<u>1</u> 0245
	24	10 <u>2</u> 45
	45	1024 <u>5</u>

e.g. 1: If the method is invoked like this:

```
System.out.println(getSmallestPair(2345));
```

the statement generates the following output:

23

e.g. 2: If the method is invoked like this:

```
System.out.println(getSmallestPair(10245));
```

the statement generates the following output:

2

e.g. 2: If the method is invoked like this:

```
System.out.println(getSmallestPair(1));
```

the statement generates the following output:

-1

Note: The input parameter does not have 2 consecutive digits.

Note: **DO NOT HARDCODE** your methods. We will test your implemented methods using a different `main` method in which different values will be passed into these methods.

---

**IMPORTANT:** Keep **Q1.java**  
in the folder <your email ID>\labtest1\Q1  
**Only this file will be marked**

---

**Question 2****[ 12 marks ]**

Find the file **Q2.java** in the folder `<your email ID>\labtest1\Q2\`. Insert your name and email ID to the top of Q2.java.

You are given the API documentation of the following Java classes and their byte code (.class) files.

(The API documentation is located at the folder `<your email ID>\labtest1\API\` and the class files are inside `<your email ID>\labtest1\Q2\`). Study the API documentation of these classes.

- Resolution
- Picture
- Album

There is a `main` method in Q2.java that has been written for you. This `main` method is for testing purposes only. Follow the instructions in this method to test your code.

Your job is to implement the following five static methods.

- `insertPicture` [2 marks]
- `printNumberOfLargePictures` [2 marks]
- `printPictureInfo` [2 marks]
- `printNumberOfPicturesByCategory` [3 marks]
- `countPicturesTakenIn` [3 marks]

Descriptions and requirements of the five methods are given in Q2.java. Strictly follow the requirements stated in Q2.java. **DO NOT change the signature (parameter and return type) of the methods.**

A sample run of Q2 when all five methods are correctly implemented are shown below.

```
There are now 6 pictures added.
```

```
There are 5 pictures with at least 4000 pixels.
```

```
The picture titled Hong Kong Airport was taken on 12/10/2013 and has 18000 pixels.
```

```
The picture titled My Birthday cannot be found in the album titled My Photo Album.
```

```
Enter picture category (p, t, s or o): p
```

```
There are 3 pictures of the category party in this album.
```

```
There are 2 pictures taken in 2013.
```

Note: **DO NOT HARDCODE** your methods. We will test your implemented methods using a different `main` method in which different values will be passed into these methods.

---

**IMPORTANT:** Keep **Q2.java**  
in the folder `<your email ID>\labtest1\Q2`  
**Only this file will be marked.**

---