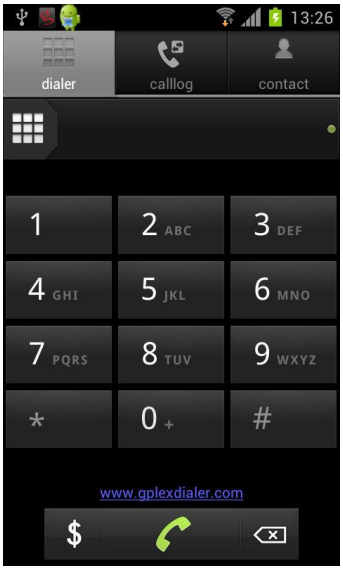
LAB 4 Selection



Work with your pair programming partner to develop an algorithm for a Java program that takes as input a single letter and displays the corresponding digit on the telephone. The letters and digits on a telephone are grouped this way:

2 = ABC	3 = DEF	4 = GHI	5 = JKL
6 = MNO	7 = PQRS	8 = TUV	9 = WXYZ

Prompt the user with an informative message for the input value like this:

```
Enter a single letter, and I will tell you what the
corresponding digit is on the telephone
R  (user enters an R)
```

The application should include the input letter as part of the output. The output that is displayed should look like this:

```
The digit 7 corresponds to the letter R on the telephone.
```

The program should display a message indicating that there is no matching digit for any non-alphabetic character entered by the user. Remember you can access the first character in an input string by scnr.next().charAt(0). Also, the application should be able to handle the case if a user enters a lower case letter. That letter character will be converted to an upper case letter character using the Character.toUpperCase(char) method.

1. Write the algorithm to solve this problem, including steps for prompting the user, getting input from the keyboard, and displaying the results.
2. Generate test cases as part of a test plan to exercise all branches of your program. Use the Test Plan document to list all of your test data and the expected results. You do not need to test every letter, however you do need to use input data that will test each branch of the selection statements; for example, the letter R tests option 7, so you do not need to test letters P, Q, or S. Show your algorithm and test plan to your TA.
3. Once your algorithm and test play have been approved, you will write the Java program, called **PhoneSelection.java**, that you designed, using your algorithm as a guide. Place your algorithm as comment blocks before the code that executes each step.
4. Place a comment block at the top of the Java program that includes the names of both programmers in the pair and course section number.
5. Use proper indentation, and meaningful identifiers throughout the code.
6. Run each of the test cases that you created and record the results on the testing document.

7. Once you are done testing your code, upload the PhoneSelection.java file to Gradescope. Check for the output of the test cases run there. Keep adjusting your code until all test cases pass.
8. You will be graded on the passed test cases for the code portion of this lab.