Practice 5

1. **Sum of digits** Write a method that computes the sum of the digits in an integer. Use the following function header:

   ```
   public static int sumDigits(int n)
   ```

   Write a test program that prompts the user to enter an integer and displays the sum of all its digits.

2. **Area of pentagon** The area of a pentagon can be computed using the following formula, in which s is the length of the sides:

   $$Area = \frac{5 \times s^2}{4 \times tan(\frac{\pi}{5})}$$

   Write a method that returns the area of a pentagon using the following header:

   ```
   public static double areaPentagon(double side)
   ```

   Write a main method that prompts the user to enter the side of a pentagon and display its area. Here is a sample run:

   ```
   Enter the side: 5.5
   The area of the pentagon is 52.04444136781625
   ```

3. **Display pattern** Write a method to display a pattern as follows:

   ```
   1

   1 2

   1 2 3

   …

   1 2 3 … (n - 1) n
   ```

   The method header should be

   ```
   public static void displayPattern(int n)
   ```

4. **Palindromes** Write a program that includes the following functions:

   a. A method with the following header that returns an integer in the reverse order:

      ```
      public static int reverseInt(int number)
      ```

      For example, `reverseInt(345)` should return `543`

b. A method with the following header that checks if an integer is a palindrome, i.e. if the integer is equal to its reverse:

```
public static boolean isPalindrome(int number)
```

For example, `isPalindrome(282)` should return `true`

Use the `reverseInt` method to implement `isPalindrome`. Also write a main method that prompts the user to enter an integer and reports whether the integer is a palindrome.

5. **Phone keypad** A phone keypad looks like the following:

Write a function called `getNumber` that takes a lowercase or uppercase letter as a parameter and returns the corresponding number. Also write a main method that prompts the user to enter a phone number as a string that may contain letters. The program should translate all (uppercase and lowercase) letters to digits, using `getNumber`, and leave all other characters intact. Here are some sample runs:

```
Enter a string: 1-800-Flowers
1-800-3569377
```

```
Enter a string: 1800flowers
18003569377
```