CSC260 Homework 3 (CH5 & CH6)

- 50 points in total
  - 10 points from MyProgrammingLab
  - 20 points from class activities
  - 20 points from Programming questions.
- Submit your homework at Canvas/Assignments/HW3
- If you have any questions about the homework, please send me an email (start with CSC260 with a section number) or open a Discussion on Canvas.

3. Programming (20 points)

1. You should make the program that returns correct answer.
2. You should print out the results and paste them as a comment.
3. For getting outputs, use the inputs from the sample run. Some programs don't need an input, then print out and copy the results.
4. Copy only Java files for submission; copy only the Java files in the `programming` directory.
5. Students earn 100% when they get correct answers and copied results, 60% when they get wrong answers, 0% when they can't compile the Java source or no answers copied.

3.1 Summation.java (5 points)

Write a program to compute the following summation.

$$sum(n) = \begin{cases} 1 + \frac{1}{1+\sqrt{2}} + \frac{1}{\sqrt{2}+\sqrt{3}} + ... + \frac{1}{\sqrt{n-1}+\sqrt{n}} & \text{if } n \geq 2 \\ 1 & \text{if } n = 1 \end{cases}$$

```
n? 1
The result: 1.0

n? 100
The result: 10.000000000000004
```

Code Hints You can see a pattern when we rewrite the experssion this way.

$$\frac{1}{\sqrt{0}+\sqrt{1}} + \frac{1}{\sqrt{1}+\sqrt{2}} + \frac{1}{\sqrt{2}+\sqrt{3}} + ... + \frac{1}{\sqrt{n-1}+\sqrt{n}}$$

So, you can translate the expression into java something like this:

```
for (int i = 1; i <= n; i++) {
    result += 1.0 / (Math.sqrt(i-1) + Math.sqrt(i));
}
```

3.2 Isbn.java (5 points)

ISBN-13 is a new standard for indentifying books. It uses 13 digits $d_1d_2d_3d_4d_5d_6d_7d_8d_9d_{10}d_{11}d_{12}d_{13}$. The last digit d_{13} is a checksum, which is calculated from the other digits using the following formula: $10 - ((d_1 + 3d_2 + d_3 + 3d_4 + d_5 + 3d_6 + d_7 + 3d_8 + d_9 + 3d_{10} + d_{11} + 3d_{12})\%10)$. In this program, get the 12-digit from the user and calculate the 13th digit. Then, print out the full 13-digit ISBN number to the screen. Following are the hints to solve this problem.

1. Create a static `int getChecksum(String s)` method.
2. The method to get the length of a string is 'length()'.
3. The method to get the i^{th} element of a string is `charAt(i)`.
4. You can use % operator to check if the location of a character is in odd or even.
5. You can change from a character '1' to a number 1 using (s.charAt(i) - '0')
6. When there is an error, you can invoke `System.exit()` method to quit the program.
7. Scanner class has a method nextLine() to get a string type input from a user.

```
Enter the first 12-digit of an ISBN number as a string: 978013213080
The ISBN number is 9780132130806

Enter the first 12 digits of an ISBN-13 as a string: 97801320
97801320 is an invalid input
```

Code hints

```
for (ALL STRINGS with i as an index);
    if (i % 2 == 0) // odd
        sum += (s.charAt(i) - '0');
    else // even
        sum += (s.charAt(i) - '0') * 3;
int checksum = 10 - sum % 10;
```

3.3 CheckPassword.java (10 points)

Some websites impose certain rules for passwords. Write a method that checks whether a string is a valid password. Suppose the password rules are as follows:

- A password must have at least eight characters.
- A password consists of only letters and digits.
- A password must contain at least two digits.

Write a program that prompts the user to enter a password and displays Valid Password if the rules are followed or Invalid Password otherwise. Make a static method `boolean isValidPassword(String s)` that checks the validity of an input.

```
Enter a string for password: Hello1234
Valid Password

Enter a string for password: BadPassword
Invalid Password
```