Question 1 [25]

Define methods with the following signatures and logic:

`public static char getRandomCharacter(boolean useAlphabets, boolean useDigits)`
This method generates a random character and returns it. If useAlphabets is true, it uses both upper and lowercase letters in the sample pool. If useDigits is true, it uses the digits 0-9 in the sample pool. If both are true, it uses both uppercase and lowercase alphabets as well as the digits 0-9 in the sample pool. Assume that at least one of the parameters must always be true.

`public static String generatePassword(boolean useAlphabets, boolean useDigits)`
This method uses a default value of 10 and 5 for the number of alphabets and digits respectively in the randomly generated password. Assume that at least one of the parameters must always be true.

`public static String generatePassword(boolean useAlphabets, boolean useDigits, int numberOfAlphabets)`
This method uses a default value of 10 for the number of digits in the randomly generated password. Assume that numberOfAlphabets cannot be less than 10.

`public static String generatePassword(boolean useAlphabets, boolean useDigits, int numberOfAlphabets, int numberOfDigits)`
This method uses the values provided by the parameters numberOfAlphabets and numberOfDigits for the number of alphabets and digits respectively in the randomly generated password. Assume that neither of them can be less than 10.

`public static String generatePassword(int lengthOfPassword, boolean useAlphabets, boolean useDigits)`
This method uses length / 2 (rounded) for both the number of alphabets and digits respectively in the randomly generated password. Assume that length cannot be less than 10.

In your main method, call all four versions of generatePassword method with user input using Scanner class. **Modify the number of inputs you need to take based on the method you are going to call. Only a single example is given below for the third variant of generatePassword(), but you need to adjust accordingly before calling other versions of the method.**

```
Should use digits? true
Should use alphabets? true
Number of alphabets: 12
Number of digits: 10
The generated password is: x22gw44y66Y6E4s0e1W3qSz
```

Question 2 [25]

In Wyoming Limited, the starting monthly salary of an employee is BDT 20,000. Every six months, the monthly salary of the employee is incremented by 12.5% of the current monthly salary. As such, after the first year, the monthly salary of the employee will be BDT 25,313. Write a program that will calculate the monthly salary of the employee after 12 years.

Also calculate and display the number of years (rounded) the employee will have to save up to buy a car that costs BDT 2,500,000, assuming that he saves 45% of his monthly salary every month.

Output format:
```
Monthly salary after 1 year: BDT 25313
Monthly salary after 2 years: BDT 32037
Monthly salary after 3 years: BDT X1
Monthly salary after 4 years: BDT X2
…
Monthly salary after 12 years: BDT X10
```

Total number of years needed to buy a car worth BDT 2,500,000 assuming 45% of monthly salary is saved: Y years.

Question 3 [25]

Write the methods with the following signatures:

```
public static double computeAreaOfNonagon(double lengthOfSide)
public static double computeAreaOfDecagon(double lengthOfSide)
```

Area of a Nonagon with a side length of *a* is:

$$A = \frac{9}{4} a^2 \cot\left(\frac{180°}{9}\right)$$

Area of a Decagon with a side length of *a* is:

$$A = \frac{5}{2} a^2 \sqrt{5 + 2\sqrt{5}}$$

Now, in your main method, use a loop to count from 10 to 50 (inclusive). Let's call the loop variable sideLength. **In each iteration, sideLength should increment by square root of 10**. Inside the loop, print out the area of a nonagon and decagon with side length of the current value of the loop variable.

Output format:
```
Area of a Nonagon with a side length of 10 is 618.18242, and that of a Decagon is
769.42088
Area of a Nonagon with a side length of 13.16 is 1070.60293, and that of a
Decagon is 1332.52617
Area of a Nonagon with a side length of 16.32 is 1646.48189, and that of a
Decagon is 2049.29405
… (rest of the values upto 50).
```

Question 4 [25]

A book in a library can have any random number of pages between 50 and 800. Write a recursive method with the following signature:

```
public static int recursiveCountPages(int numberOfBooks)
```

You must return the total number of pages of all the books from this method. **You cannot use multiplication**. Also implement the recursive method in an iterative way using the following method signature:

```
public static int iterativeCountPages(int numberOfBooks)
```

In your main method, call the recursive method with three values **(use of Scanner class is mandatory)**. Also call the iterative method with the same three values to verify that both your solutions are working correctly.

Output format:
```
Using recursive method:
For X1 books, the total number of pages is Y1.
For X2 books, the total number of pages is Y2.
For X3 books, the total number of pages is Y3.

Using iterative method:
For X1 books, the total number of pages is Y1.
For X2 books, the total number of pages is Y2.
For X3 books, the total number of pages is Y3.
```