Practice Problem 1

Note:

1. Submit java separate **java** file for each problem.
2. File name must be **<Name_Universityrollno_programNo.java>**. For example, if **Ramesh** is submitting **6th** program and your university roll no is **2012004** then your file name must be '**Ramesh_2012004_06.java**'. If any code contains more then one java file then numbering will **<Name_ Universityrollno_programNo_1.java>** for first file **<Name_ Universityrollno_programNo_2.java>** for second and so on. Example '**Ramesh_2012004_06_1.java**' and '**Ramesh_2012004_06_2.java**'.
3. If you did not follow the naming convention then your program will not be checked and awarded 0 in the same program.
4. Every file must contain the comments at the start of program. These comments must contain your name, Date of writing the program and problem statement.
5. Please do not copy the codes from your friend's codes will be send for plagiarism check. If any code will be caught under plagiarism, then he will be awarded as 0 in complete assignment.

---

1. Java Program to Multiply two Floating Point Numbers (Numbers must be taken as command line arguments).
2. Java Program to Find GCD of two Numbers (Numbers must be taken as command line arguments).
3. Java Program to Display Armstrong Number Between Two Intervals (Numbers must be taken as command line arguments).
4. Java Program to Convert Binary Number to Decimal and vice-versa.
5. Java Program to Convert Octal Number to Decimal and vice-versa.
6. Java Program to Convert Binary Number to Octal and vice-versa.
7. Java Program to Find Largest Element of an Array
8. Java Program to Count the Number of Vowels and Consonants in a Sentence
9. Java Program to Calculate Difference Between Two Time Periods
10. Java Program to Add Two Dates
11. Java Program to Check if a String is Numeric.
12. Write a java program to take input as a command line argument. Your name, course, universityrollno and semester. Display the information.

    Name:

    UniversityRollNo:

    Course:

    Semester:

13. Using the switch statement, write a menu-driven program to calculate the maturity amount of a bank deposit.

    The user is given the following options:

(i) Term Deposit

(ii) Recurring Deposit

For option (i) accept Principal (p), rate of interest (r) and time period in years (n). Calculate and output the maturity amount (a) receivable using the formula a = p[1 + r / 100]n.

For option (ii) accept monthly instalment (p), rate of interest (r) and time period in months (n). Calculate and output the maturity amount (a) receivable using the formula a = p * n + p * n(n + 1) / 2 * r / 100 * 1 / 12.

For an incorrect option, an appropriate error message should be displayed.

[Use Scanner Class to take input]

14. Program to find if the given numbers are Friendly pair or not (Amicable or not). Friendly Pair are two or more numbers with a common abundance.

Input & Output format:

- Input consists of 2 integers.
- The first integer corresponds to number 1 and the second integer corresponds to number 2.
- If it is a Friendly Pair display Friendly Pair or displays Not Friendly Pair.
- **For example**, 6 and 28 are Friendly Pair.

(Sum of divisors of 6)/6 = (Sum of divisors of 28)/28.

Steps to check whether the given numbers are friendly pair or not.

- Input the numbers num1 and num2.
- Initialize sum1 = sum2 = 0.
- sum1 = sum of all divisors of num1.
- sum2 = sum of all divisors of num2.
- If (sum1 == num1) and (sum2 == num2), then print "Abundant Numbers".
- Else, print "Not Abundant Numbers".

Program to check whether the given numbers are friendly pair or not.

15. Program to replace all 0's with 1 in a given integer. Given an integer as an input, all the 0's in the number has to be replaced with 1.

For example, consider the following number:

Input: 102405

Output: 112415

Input: 56004

Output: 56114

Steps to replace all 0's with 1 in a given integer:

- Input the integer from the user.

- Traverse the integer digit by digit.
- If a '0' is encountered, replace it by '1'.
- Print the integer.