Create a class YourName.java and only upload this java file in LEA. Each question has 6 marks.

1. Implement each question in a separate method or class.
2. Add comments to your code.
3. Use meaningful names for variables and classes.
4. Pay attention to indentation.
5. Remove extra lines.
6. Respect Java conventions.
7. Your code must be easy to read, easy to maintain and reusable.
8. Test your code by calling your methods from main method.
9. You have to be able to explain your code and answer my questions.

1- Implement userInput method to receive different type of inputs from user.
   1.1 At the beginning application will ask user about input type and number of inputs user wants to have.
   1.2 User will be able to terminate the execution by typing “exit”
   1.3 Application will ask user if the user wants to enter more data or finish the program. Y or y means continue. Any other answer will finish the program.
   1.4 At the end application will show user inputs.
   1.5 Implement main method and test your application.

```
Please select your input type:
1- for int     2- for double     3- for String
1
Please define number of inputs: 3
Please enter an int: 1
Please enter an int: 2
Please enter an int: exit
Thank you
[1, 2, 0]
Do you want to continue: Y  N y

Please select your input type:
1- for int     2- for double     3- for String
1
Please define number of inputs: 3
Please enter an int: 1
Please enter an int: 2
Please enter an int: 3
[1, 2, 3]
Do you want to continue: Y  N n

Process finished with exit code 0
```

2- (Perfect number) A positive integer is called a perfect number if it is equal to the sum of all of its positive divisors, excluding itself. For example, 6 is the first perfect number because 6 = 3 + 2 + 1. The next is 28 = 14 + 7 + 4 + 2 + 1. There are four perfect numbers less than 10,000. Write a program to find all these four numbers.

3- (Longest common prefix) Write a program that prompts the user to enter two strings and displays the largest common prefix of the two strings. Here are some sample runs:

```
Enter the first string: Welcome to C++

Enter the second string: Welcome to programming The common
prefix is Welcome to

Enter the first string: Atlanta

Enter the second string: Macon Atlanta and Macon have no
common prefix
```

4- (Palindrome integer) Write the methods with the following headers

```
// Return the reversal of an integer, i.e., reverse(456)
returns 654 public static int reverse(int number)

// Return true if number is a palindrome public static
boolean isPalindrome(int number)
```

5- (Check password) 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.

Optional (Extra Marks)

6- (Game: hangman) Write a hangman game that randomly generates a word and prompts the user to guess one letter at a time, as shown in the sample run. Each letter in the word is displayed as an asterisk. When the user makes a correct guess, the actual letter is then displayed. When the user finishes a word, display the number of misses and ask the user whether to continue to play with another word. Declare an array to store words, as follows:

```
// Add any words you wish in this array String[] words =
{"write", "that", ...};

(Guess) Enter a letter in word ******* > p
(Guess) Enter a letter in word p****** > r
(Guess) Enter a letter in word pr**r** > p

p is already in the word

(Guess) Enter a letter in word pr**r** > o
(Guess) Enter a letter in word pro*r** > g
(Guess) Enter a letter in word progr** > n

n is not in the word

(Guess) Enter a letter in word progr** > m
(Guess) Enter a letter in word progr*m > a


The word is program. You missed 1 time Do you want to guess
another word? Enter y or n>
```