Question - 1
Java-Movie Analysis

Create a class DC with the below attributes:

movieId - int
movieName - String
budget - double (in crores)
boxOffice - double (Box Office Collection in crores)
rating-int

The above attributes should be private, write getters, setters, and parameterized constructors as required.

Create a class Solution with the main method.

Implement two static methods - findAverageBudget and searchMovieByBoxOffice in the Solution class.

findAverageBudget method:
This method will take one input parameter - an array of DC objects.
The method will return the average budget of movies from an array of DC objects if the average budget of the movies is greater than 0. Else the method should return 0.

searchMovieByBoxOffice method:
This method will take two input parameters, an array of DC objects as one parameter and a double value as the another parameter.
The method will return DC object array in descending order of their movieId, from the array of DC objects whose Box Office collection is less than the given Box Office Collection (double parameter passed). If no DC movies with the given condition are present in the array of DC objects, then the method should return null.

Note:
1. No two DC objects would have the same movieId.
2. All the searches should be case insensitive.

The above-mentioned static methods should be called from the main method.

For findAverageBudget method - The main method should print the returned average budget as it is, if the returned value is greater than 0, or it should print "No Budget".

For searchMovieByBoxOffice method - The main method should print the movieId from the returned DC object array if the returned value is not null.
If the returned value is null then it should print "No such movies".

Before calling these static methods in the main, use the Scanner object to read the values of four DC objects referring to attributes in the above-mentioned attribute sequence. Next read one double parameter for capturing box office.

Consider below sample input and output:
Testcase1:
Input:
1001
BatMan
600.0
800.0
4
1002

```
Adam
700.0
900.0
3
1005
WonderWomen
1200.0
3700.0
5
1003
JusticeLeague
1500.0
3000.0
5
5000
```

Output:

```
1000.0
1005
1003
1002
1001
```

Testcase2:

Input:

```
1001
BatMan
600.0
800.0
4
1002
Adam
700.0
900.0
3
1005
WonderWomen
1200.0
3700.0
5
1003
JusticeLeague
1500.0
3000.0
5
100
```

Output:

```
1000.0
No such movies
```

****

Sample code snippet for reference:

Please use the below code to build your Solution.

```
import java.util.Scanner;
public class Solution
{
  public static void main(String[] args)
 {
  //code to read values
  //code to call required method
  //code to display the result
 }

 //code the first method


 //code the second method
}

//code the class
------------------------------------------------
```

Note on using Scanner object:
Sometimes scanner does not read the new line character while invoking methods like nextInt(), nextDouble() etc.
Usually, this is not an issue, but this may be visible while calling nextLine() immediately after those methods.
Consider below input values:
1001
Savings
Referring below code:
Scanner sc = new Scanner(System.in);
int x = sc.nextInt();
String str = sc.nextLine(); -> here we expect str to have value Savings.Instead it may be "".
If the above issue is observed, then it is suggested to add one more explicit call to nextLine() after reading the numeric value.