CS526 O2
Homework Assignment 1

Problem 1 (50 points)

This problem is a practice of using basic Java features, including selection statements, iteration statements, and using arrays.

An incomplete program named *Hw1_p1.java*, which must include three methods, is provided. You must implement two methods within the program as described below. The third method, which is a *main* method, is already written in the program and you can use it to test your program.

The two methods you must implement are:

- *find* method
  - Signature:

    ```
    public static void find(int[] a, int x)
    ```

  - Behavior:
    - Receives an array of integers *a* and an integer *x*.
    - Performs a linear search on *a* for *x*.
    - If *x* is in *a*, print the indexes of the array slots where *x* is stored (there may be multiple *x*'s in *a*) on the screen.
    - If *x* is not in *a*, print the message "*x* does not exist" on the screen.

  Note that you should not use Java's built-in method. Your program must scan and search the array.

- *isPrifix* method
  - Signature:

    ```
    public static boolean isPrefix(String s1, String s2)
    ```

  - Behavior:
    - Receives two strings *s*1 and *s*2. Assume that length of *s*1 <= length of *s*2.
    - Returns true if *s*1 is a prefix of *s*2 and returns false otherwise.

  Note that you should not use Java's built-in method. Your program must determine whether *s*1 is a prefix of *s*2 by comparing characters, one at a time, in the two strings.

- *main* method
  The main method is already written in the given file and it is used to test above two methods. After completing the above two methods, if you run your program, your output must be:

```
5 is in a[0]
5 is in a[2]
5 is in a[7]
10 does not exist

abc is a prefix of abcde
abc is not a prefix of abdef
```

You may want to test your program with different input arguments.

Problem 2 (50 points)

The goals of this problem are:

- Practice of simple I/O
- Creating and using Java objects
- Storing and accessing Java objects in an array

The definition of a *Car* class is given in the *Car.java* file. Note that you must use the given *Car* class that is posted with Homework 1 (do not use the *Car* class that was used in Chapter 1 lecture).

You must write a program named *Hw1_p2.java* that satisfies the following requirements:

- Your program must have three methods – f*indByMake*, *newerThan*, and *main* methods.

- *findByMake* method
  - Signature

    ```
    public static void findByMake(Car[] cars, String make)
    ```

  - Behavior
    - Receives an array of *Car* objects and the make of a car *make*.
    - Finds and prints all cars in the array that have the same make as the given *make*.
    - Prints an appropriate message, if there is no car with the given *make*.
    - No return value.

- *newerThan* method
  - Signature:

    ```
    public static void newerThan(Car[] cars, int year)
    ```

  - Behavior
    - Receives an array of *Car* objects and the year of a car *year*.
    - Finds and prints all cars in the array that are newer than the given *year* (in other words, the cars made later than the given *year*).
    - Prints an appropriate message, if there is no car newer than the given *year*.

    - No return value

- *main* method
    - The program must read information about cars from an input file named *car_input.txt*.
    - There are 10 lines in the input file and it has one car information in each line, as shown below:

```
GM, 20000, 2014
Honda, 18000, 2014
Hyundai, 25000, 2020
Kia, 15000, 2010
Ford, 35000, 2021
Toyota, 30000, 2020
Ford, 17000, 2010
Honda, 20000, 2016
Chevy, 25000, 2019
Volvo, 30000, 2021
```

    - Your program must create an array of *Car* objects, *cars*, of size 10.
    - Your program must read the input file, one at a time.
    - After reading each line, your program must create a *Car* object and add it to the array *cars*.
    - Your program must print all cars in the array on the screen.
    - Your program must invoke the *findByName* method.
    - Your program must invoke the *newerThan* method.

An incomplete *Hw1_p2.java* file is provided, where an incomplete *main* method is included. You must complete the program.

If you run the program and pass Honda and 2017 as parameters to the two methods, the following is an expected output:

```
All cars:
       Make = GM      Year = 2014    Price = 20000
       Make = Honda   Year = 2014    Price = 18000
       Make = Hyundai Year = 2020    Price = 25000
       Make = Kia     Year = 2010    Price = 15000
       Make = Ford    Year = 2021    Price = 35000
       Make = Toyota  Year = 2018    Price = 30000
       Make = Ford    Year = 2010    Price = 17000
       Make = Honda   Year = 2016    Price = 20000
       Make = Chevy   Year = 2019    Price = 25000
       Make = Volvo   Year = 2015    Price = 23000

All cars made by Honda
       Make = Honda   Year = 2014    Price = 18000
       Make = Honda   Year = 2016    Price = 20000

All cars made after 2017
       Make = Hyundai Year = 2020    Price = 25000
       Make = Ford    Year = 2021    Price = 35000
       Make = Toyota  Year = 2018    Price = 30000
       Make = Chevy   Year = 2019    Price = 25000
```

Grading

Problem 1:

- If your program does not compile, 40 points are deducted.
- If your program compiles but causes a runtime error, 30 points are deducted.
- For each method, if there is no output or output is completely wrong, 10 points are deducted.
- For each method, if your output is partially wrong, up to 10 points are deducted.

Problem 2:

- If your program does not compile, 40 points are deducted.
- If your program compiles but causes a runtime error, 30 points are deducted.
- For each method, if there is no output or output is completely wrong, 10 points are deducted.
- For each method, if your output is partially wrong, up to 10 points are deducted.

Deliverable

No separate documentation is needed. However, you must include the following in your source code:

- Include the following comments above each method:
  - Brief description of the method
  - Input arguments
  - Output arguments
- Include sufficient inline comments within your source code to increase readability of your code and to help readers better understand your code.

You must submit the following files:

- *Hw1_p1.java*
- *Hw1_p2.java*