INFO 2313: Assignment–3

School of Business
Kwantlen Polytechnic University, Canada
`https://courses.kpu.ca/course/view.php?id=6373`

Instructor: Ali Madooei
`ali.madooei@kpu.ca`

June 24, 2016

$\boxed{\textbf{It is due at 12:00 pm (noon) on Friday, July 1, 2016}}$

This assignment is covering the material in Chapters 6-8 [Liang, 2015] with focus on using arrays and implementing methods. It contains three tasks and carries 20 marks.

Guidelines

1. Create a new Java project and name it `assign03`
2. Create the following Java classes: `Task01`, `Task02`, `Task03`; make sure these classes contain `public static void main(String[] args)` method.
3. Attempt to implement each task and do your best. Just as there are different ways to say the same thing in English, there are different ways to do the same thing with a programming language. Any solution is acceptable as long as the code is error-free, implements what I have asked for, and produces the correct output.
4. I encourage you to follow good programming practice (e.g. include comments, code indentation, etc). This practice will be considered in marking.

Task 1. Consider the following code:

```
public class Task01 {

        public static int[] find (int[] A, int target)
        {
                int [] results = new int[A.length];

                /*
                *   Your code goes here
                */

                return results;
        }

        public static void main (String[] args)
        {
                int[] A = {5,0,1,5,3,4,5,3,5};

                int[] B = find(A, 5);
                System.out.println("B = " + Arrays.toString(B));

                int[] C = find(A, 2);
                System.out.println("C = " + Arrays.toString(C));
        }

}
```

complete the implementation of `find` method such that if you run the code, it would produce the following output:

```
B = [1, 0, 0, 1, 0, 0, 1, 0, 1]
C = [0, 0, 0, 0, 0, 0, 0, 0, 0]
```

Task 2. Rolling a die multiple times and analyzing the outcome is a common example in understanding probability and statistics. Write a Java method that rolls a die n times and prints a histogram like the one shown below (for $n = 100$). **Hint:** Use Java's `Math.random()` method to simulate rolling a die.

```
[1] : ****************
[2] : ******************
[3] : ****************
[4] : *******************
[5] : ****************
[6] : ***************
```

In the histogram above, numbers 1 to 6 represent each side of a die (Fig-1). The total number of asterisks on each line shows the number of times die landed on that number. For instance, there are 10 asterisks in front of `[1]` which indicates that in my simulation of rolling a die for 100 times, the face 1 came up 10 times.

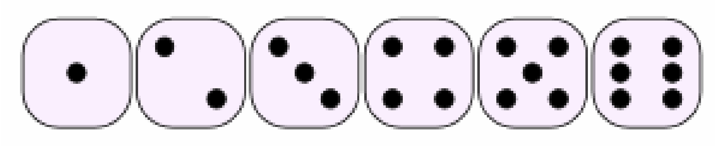

Figure 1: The common die has six faces.
We usually call the faces 1, 2, 3, 4, 5 and 6

The method header should look like:

```
public static void die_experiment (int n)
```

Call `die_experiment` in the main method with different values for `n`.

Task 3. Create a Java method

```
public static int[] multPoly (int[] A, int[] B)
```

which computes $C = A \times B$ where `A` and `B` are two 1D array of size `n` and `C` is a 1D array of size $2n-1$. The arrays A and B contain the coefficients of $(n-1)$-degree polynomials. Your function should implement Algorithm-1.

Algorithm 1 – Multiply two polynomials

```
    Input: Two arrays A and B of size n.
    Output: Array C of size 2n-1 where $C = A \times B$.
 1: Create a 2D array pairs[n][n]
 2: for i=0 to n-1 do
 3:     for j=0 to n-1 do
 4:         pairs[i][j]=A[i]×B[j]
 5:     end for
 6: end for
 7: Create a 1D array C[2n-1]
 8: Initialize all elements of C to zero
 9: for i=0 to n-1 do
10:     for j=0 to n-1 do
11:         C[i+j]=C[i+j]+pairs[i][j]
12:     end for
13: end for
14: Return C
```

Generally, an $(n-1)^{\text{th}}$ order polynomial in variable x is written as

$$P(x) = a_{n-1}x^{n-1} + a_{n-2}x^{n-2} + \cdots + a_1 x + a_0$$

It is natural to associate an array A with P that contains the coefficients:

$$A = [a_{n-1},\ a_{n-2},\ \ldots a_1,\ a_0]$$

The array A can be used to uniquely identify the polynomial P. If you have forgotten about polynomial multiplication, take a look at the footnote[1] link. To test your function, write a simple test driver (in the main method):

- Ask user to enter n
- Ask user to enter the coefficients of $A(x)$ and $B(x)$ and read them into `A` and `B`
- Call `multPoly` function to compute $C = A \times B$.
- Display the coefficients in C.

[1] `http://www.wallace.ccfaculty.org/book/5.5%20Multiply%20Polynomials.pdf`

As an example, consider multiplying $A(x) = 4x^3+3x^2+2x+1$ and $B(x) = x^3+2x^2+3x+4$. The resulting polynomial is $C(x) = 4x^6 + 11x^5 + 20x^4 + 30x^3 + 20x^2 + 11x + 4$. The following sample output shows how your program must work.

```
Enter n: 4
Enter coefficients of A
4 3 2 1
Enter coefficients of B
1 2 3 4

Coefficients of C=AxB are [4, 11, 20, 30, 20, 11, 4]
```

In cases where two polynomials are of different order, you can put zero for the missing terms. For example, to multiply $A(x) = 4x^3$ with $B(x) = 5x^2 - 2x + 5$ (result is $20x^5 - 8x^4 + 20x^3$):

```
Enter n: 4
Enter coefficients of A
4 0 0 0
Enter coefficients of B
0 5 -2 5

Coefficients of C=AxB are [0, 20, -8, 20, 0, 0, 0]
```

Submission

Once you are done with your solutions, please include the following comment block (with you name and student number) on top of every Java file in your project.

```
/*
*  INFO 2313 - SUMMER 2016
*  ASSIGNMENT 3
*
*  Student Name:
*  Student Number:
*/
```

Then, zip the project `assign03` folder and upload it to Moodle.