Problem 1:

```
The result of test case 1 is:
3 7 9

The result of test case 2 is:
1 2 7 9

The result of test case 3 is:
1 1 7 7 10 10 15 18

The result of test case 4 is:
1 3 5 5 5 5 6 8 10 12 15 16 17 17 18 20 21 25 28
```

Problem 2 AlaCarte Multiplication:

```
AlaCarte Multiplication

The result of 7000 * 7294 = 51058000

The result of 25 * 5038385 = 120921240

The result of -59724 * 783 = -46763892

The result of 8516 * -82147953548159344 = -699571972416124973504

The result of 45952456856498465985 * 98654651986546519856 = 4533423639104649634298438798517796578304

The result of -45952456856498465985 * -98654651986546519856 = 4533423639104649634298438798517796578304
```

Problem 2 Rectangle Multiplication:

```
Rectangular Multiplication

The result of 7000 * 7294 = 51058000

The result of 25 * 5038385 = 125959625

The result of -59724 * 783 = -46763892

The result of 8516 * -82147953548159344 = -699571972416124973504

The result of 45952456856498465985 * 98654651986546519856 = 4533423639104649634397093450504343098160

The result of -45952456856498465985 * -98654651986546519856 = 4533423639104649634397093450504343098160
```

Statement:
My platform uses 64 bit CPU (intel Core i5-8300H), the programming language is Java, and the code is compiled on IntelliJ IDEA Community Edition 2020.3.
Some numbers in the test cases, like: -82147953548159344, have too many digits so the int and long type variables cannot handle these numbers. Therefore, I use

java.math.BigInteger to handle extremely big or small number in test cases.
The BigInteger is only used to show the extremely big and small numbers. All the functions of BigInteger I use can be replaced by operator + - * / % < > and = if the variables are int or long (The BigInteger cannot use these normal operators, they have to use functions like "BigInteger.add()", ".multiply()" to execute operation). None of them have other functions.
Just in case using BigInteger is not allowed in this project, I also submitted normal version of AlaCarte and Rectangular, which don't use BigInteger. So if using java.math.BigInteger is not allowed, please grade the normal version programs. However, some test cases that have extremely big or small number cannot be tested in normal version because the expression range of int and long variables is not big enough in java.
The normal version codes are in the folder: "Normal_version". If using BigInteger is allowed, please ignore normal version. Thank you! :)