ICS4U Basic Methods

Marks: 25

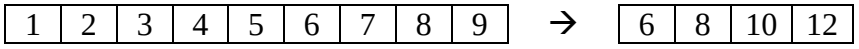
For each of the following questions write a program to test your Methods.

1. Write a Method called **mean** that takes an array of integers, and **returns** the mean of all the numbers as a double. Save as **Method1.java** (2 marks)

2. Write a Method called **repeat** that takes a string, and an integer then outputs the string on the screen a number of times indicated by the integer. Save as **Method2.java** (2 marks)

3. Write a Method called **arrayAdd()** that takes two parameters, an array of ints and a single int. Your method will **return a new array** with the second parameter added to the end of the array. Save as **Method3.java** (4 marks)

4. Write a Method called **fold** that takes an array of integers and returns an array that is half the original size (rounded down.) The new array will be computed by adding the first half of the array with the second half of the array. E.g.



Save as **Method4.java** (5 marks)

5. Write a Method called **licensePlate** that takes an array of objectionable words and returns a random license plate that does not have any of those words in it. Your plate should consist of four letters a space then three numbers. You may use placeholder words to test your program. If you decide to use real words to test your program then make sure you don't leave your source code lying around. Save as **Method5.java** (6 marks)

6. I've been playing around with a Remote-Controlled tank in the field beside my house. The mini-rocket launcher is cool and all, but after hours of navigating terrain and blowing stuff up I came to realize that navigating is really boring (watching little plastic men melt never gets old on the other hand.) I want you to make a recursive method to do the navigation for me. I measured out an 8m x 8m section of the field and calculated how long the tank takes to navigate from one corner to the opposite one. For the sake of simplicity I only need the Tank to go down and to the right. Fortunately my tank turns almost instantly.

Input and Output

The method will take as a parameter an 8 x 8 grid of numbers starting with the top left corner (0,0) and ending with the bottom right(7,7). These numbers are the number of seconds that it takes for the tank to move onto that particular square (oddly enough it takes the same time regardless of the direction it comes from.) Your output will be the fastest time the tank can make it to the last square as well as proper directions. You may assume that it will always take less than 1000 seconds total.

Sample Input

0	12	7	43	32	12	30	15
21	26	18	34	41	9	17	21
20	43	23	35	23	20	17	37
5	29	28	18	9	42	35	24
25	15	36	25	21	9	14	19
25	26	32	18	17	19	25	15
35	15	12	21	24	26	14	35
15	12	18	25	14	22	21	15

Sample Output

234

RRDDDRRDRRDDDR

Note: When you declare your 2D array, you need to transpose it, or you will be dealing in y,x coordinates.

e.g.

```
int [][]grid =
{{ 0,21,20, 5,25,25,35,15},
{12,26,43,29,15,26,15,12},
{ 7,18,23,28,36,32,12,18},
{43,34,35,18,25,18,21,25},
{32,41,23, 9,21,17,24,14},
{12, 9,20,42, 9,19,26,22},
{30,17,17,35,14,25,14,21},
{15,21,37,24,19,15,35,15}};
```

Save as **Method6.java** (6 marks)