Montgomery College
2019 Programming Competition
Intro Teams

Scoring:

- Each problem is worth of 10 points. There are 6 problems.
- Teams are ranked based on the number of points received, the highest score first.
- Teams that tie on points will be ranked according to the number of problems solved/attempted.
- Teams that tie on both points and the number of problems solved/attempted will be ranked according to the tie-break problem #6.

Criteria	**Points Awarded**
Compiles, runs, passes all (100%) testing	**10**
Compiles, runs, passes more than 50% and less than 100% of testing	**7**
Compiles, runs, passes less than 50% of testing	**3**
Compiles, runs, passes all testing, but fails to implement the solution as specified. For example, requirements call for the use of a two-dimensional array, the submitted solution did not use a two-dimensional array. Or, the requirement asks for a recursive implementation, the submitted solution did not use recursion.	**5**
Compiles, runs, hardcodes the expected output	**0**
Does not compile	**0**

Teams are required to return this problem set to the competition staff at the end of the competition.

BasicIO Program:

```
import java.util.Scanner;
public class BasicIO {
   static Scanner stin = new Scanner (System.in);

   static boolean menu () {
      System.out.print ("Enter q to quit: ");
      String line = stin.nextLine().trim();
      if (line.length() == 0) return true;
      if (line.startsWith ("q")) return false;
      Scanner sline = new Scanner (line);
      // YOUR CODE HERE
      return true;
   } // end menu

   public static void main (String [] args) {
      System.out.println ("Problem <n>: <name>");
      while (menu ());
      System.out.println ("Bye");
   } // end main

} // end class BaseIO
```

Problem 1 – Compute a weighted average: 3 integer version

This problem has a number of steps.

a) Extend the BasicIO program to read in 3 integers and echo them back to the user and compute their sum.
b) Extends part a to read two sets of 3 integers and compute their inner product:
a b c d e f → a*d + b*e + c*f = F(a,b,c,d,e,f)
The result should be an integer.
c) Extend part b to compute the average of the a, b and c weighted by d, e and f:
W(a,b,c,d,e,f) = F(a,b,c,d,e,f) / (d + e + f)
The result should be a double. v
d)
e) NOTE: Since you are starting with BasicIO, the program should keep running until the user enters a 'q'.
f) Your program may assume that the inputs are correct in number and format.

Example of (c):

```
Problem 1: 3 integer weighted average
Enter q to quit: 1 2 3 1 1 1
Average is:  2.00
Enter q to quit: 1 2 3 4 5 6
Average is:  2.13
Enter q to quit: 13 15 17 1 2 3
Average is: 15.67
Enter q to quit: q
Bye
```

Problem 2 – Compute a weighted average: N integer version

Extend your solution to Problem 1 to use integer arrays and use the first integer as the number of integers in each of the two arrays. Note: the arrays should be declared after the first integer is read. Your program may assume that the inputs are correct in number and format.

Example:

```
Problem 2: N integer weighted average
Enter q to quit: 3 1 2 3 4 5 6
Average is:  2.13
Enter q to quit: 3 13 15 17 1 2 3
Average is: 15.67
Enter q to quit: 5 1 2 3 4 5 6 6 6 6 6
Average is:  3.00
Enter q to quit: 5 1 2 3 4 5 2 3 2 5 5 1
Average is:  3.47
Enter q to quit: 2 1 5 1 2
Average is:  3.67
Enter q to quit: 2 1 5 2 1
Average is:  2.33
Enter q to quit: q
Bye
```

Problem 3 – Modulo N multiplication table

The user will enter an integer, N, and your program will print the multiplication table modulo N.
Hints:

a) start with the BasicIO code.
b) Use the following formatting:

```
System.out.printf ("%5d", r*c%n);
```

Example:

```
Problem 3: Modulo N multiplaction table
Enter q to quit: 1
    0
Enter q to quit: 2
    0    0
    0    1
Enter q to quit: 3
    0    0    0
    0    1    2
    0    2    1
Enter q to quit: 4
    0    0    0    0
    0    1    2    3
    0    2    0    2
    0    3    2    1
Enter q to quit: 5
    0    0    0    0    0
    0    1    2    3    4
    0    2    4    1    3
    0    3    1    4    2
    0    4    3    2    1
Enter q to quit: q
Bye
```

Problem 4 – Hand of n cards

Consider a deck of standard playing cards ({A, 2, 3, 4, 5, 6, 7, 8, 9, 10, J, Q, K} in each of 4 suites {spades, hearts, diamonds, clubs}).
Write a program that will take an integer N from the user and generate a random hand of N cards. For full credit, your program should insure that cards are not repeated, as if the cards were dealt from the deck.
Your program should be based on the BasicIO program provided, and continue asking for input until the user enters a "q".
Note: The suit symbols can be represented in Java as Unicode String literals (spades, hearts, diamonds, clubs):

```
static String [] s = {"\u2660", "\u2665", "\u2666", "\u2663"};
```

Partial Credit:

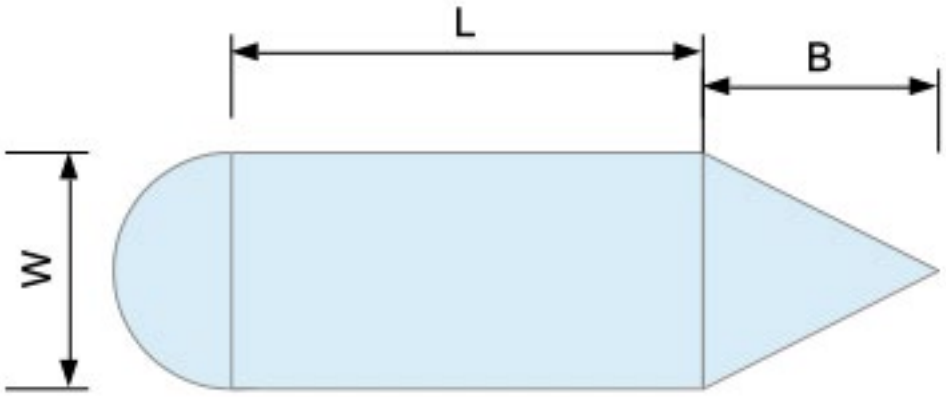
- For 7 points – create a program that allows repeated cards in the hand.
- For 10 point – make sure that each card only appears once.

Example:

```
Problem 4: Hand of cards
Enter n or q to quit: 5
A♣ 6♥ 9♦ 6♣ K♠
Enter n or q to quit: 3
Q♦ 9♥ 2♦
Enter ns or q to quit: 13
4♥ J♣ 2♥ 3♠ A♣ 7♦ 10♥ K♥ 6♣ 10♣ 10♦ K♣ Q♣
Enter n a b or q to quit: q
Bye
```

Problem 5 – Area of a figure

Write a program that will accept 3 double values from the user, W, L and B, and compute the area of the figure below. Your program should be based on the BasicIO program provided, and continue asking for input until the user enters a "q".



Example:

```
Problem 5: Area of a figure
Enter W L and B or q to quit: 1 1 1
Area is: 1.8927
Enter W L and B or q to quit: 1 2 3
Area is: 4.2854
Enter W L and B or q to quit: 50 80 900
Area is: 27,481.7477
Enter W L and B or q to quit: q
Bye
```

Problem 6 (Tie-Breaker): – Simple Arithmetic

Write a program that will accept an operation and two double values and compute the result of the operation applied to the doubles.
Your solution should used the Java switch structure, and the answer should be presented with 5 digits after the decimal point.
Your program should be based on the BasicIO program above

Example:

```
Problem 6: Simple Arithmetic
Enter op {add, sub, mult, div, exp} a and b or q to quit:
add 3 4
   3.00000 + 4.00000 = 7.00000
Enter op {add, sub, mult, div, exp} a and b or q to quit:
div 5 123
   5.00000 / 123.00000 = 0.04065
Enter op {add, sub, mult, div, exp} a and b or q to quit:
exp 2.354 9.4453
   2.35400 ^ 9.44530 = 3,249.55239
Enter op {add, sub, mult, div, exp} a and b or q to quit:
were 4 5
   4.00000 illegal operator 5.00000 = 0.00000
Enter op {add, sub, mult, div, exp} a and b or q to quit:
q
Bye
```