**CSC142, Computer Science II, Project 4 assignment**

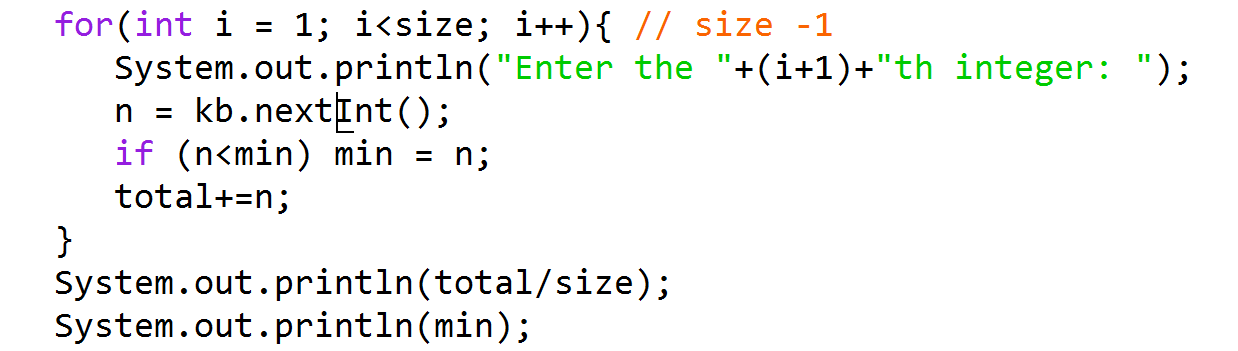
Develop the following loop programs and submit each java file to D2L. Non-loop program or later submission is not accepted.

Part 1: Basic concepts of event control (Extend the lecture samples for the solution)

1. Sum.java (10 pts): Write a program that will repeatedly prompt the user to type a number until the user types a POSITIVE number, then print the sum of the digits of that number. For instance, -1 or 0 will force the user to re-enter the number and 12345 will lead to the print of 15.
2. Perfect.java (10 pts): Write a program that reads in an integer *N* from the keyboard, and displays whether *N* is a "perfect number" or not (Yes or No). A number is "perfect" if it is equal to the sum of all of its factors (not including itself as a factor, but including 1 as a factor). 6 is the first perfect number, because its factors are 1, 2, and 3, and 1+2+3 = 6.

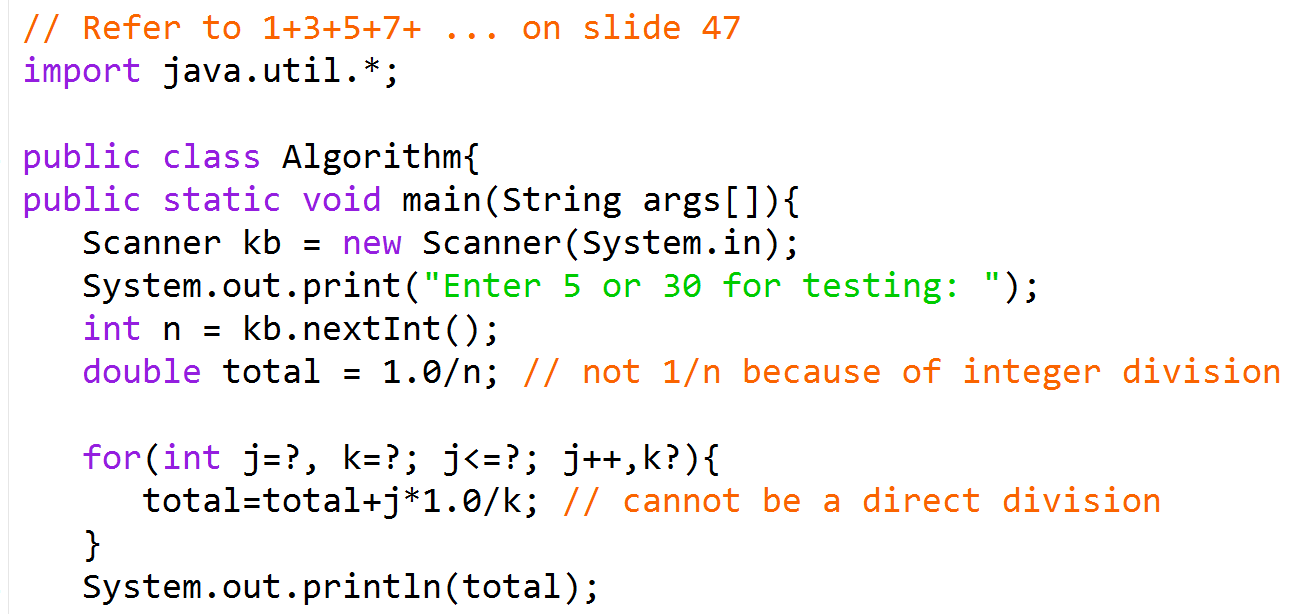
Part 2: Advanced development (in complicated situations).

1. MinAvg.java (10 pts): Write a program that reads 500 integers and displays its minimum and average. Please test your code with 5 integers first and consider the extreme case when all integers are positive. For instance, by given 1, 2, …, 5, the minimum will be 1. Hint: the comparison starts from the second input number, not applying on the first one. So, set the target “min” with the first input number. Then, implement a loop with (500-1) or (5-1) rounds to update this minimum record if we have a new input to win (i.e., <). A sample of loop is provided here. Please figure out what must be initialized.



1. Algorithm.java (30 pts): Workbench 5, page 260, calculate 1/30+2/29+3/28+…+30/1. Your program should read the number 30 from keyboard. If this input value changes, your program should also obtain the correct result. Be careful of the need for the casting! 1/30 is 0 in java unless (double)1/30 is needed. Suggest everyone to develop the iteration body from a similar process 1+2+3+…+30 and test with 2, 3, and 4. The results must be 1/2+2/1=2.5, 1/3+2/2+3/1=4.33333, and 1/4+2/3+3/2+4/1=6.41667.

Step 1) Your program should adopt the pattern in the picture:



Step 2) To complete both 1/30+2/29+…+30/1 and 1/5+2/4+…+5/1 with the same code, what will be the value of dividend j in the last number “j\*1.0/k” added into total? How to use an expression to express this 30 and 5? Use this expression for the test part “j<=?;”

Step 3) To complete 1/30+2/29+, what will be the computer’s first iteration if total has been initiated as “1.0/n”, 1/30 or 2/29? Step 3) What will be the value of dividend j in the first iteration “total = total + j\*1.0/k”? Use this value in the for-loop initialization: “int j=?,” What will be the value of divisor k in the first iteration “total = total + j\*1.0/k”? Use this value (not “n”!) in the for-loop initialization: “k=?;”

Step 4) If the dividend j is increased, what about the advisor k? Implement the update of k in “k?”

Step 5) Test the complete version of your code before the submission.

1. RC.java (30 pts): Write code that reads in two numbers C and R from the keyboard, for instance, if the user enters a 5 for C and 4 for R and C>R, and displays a figure with **R** rows and **(2\*C-1)**-columns of "$" characters:

$$ $$

$$$ $$$

$$$$ $$$$

$$$$$$$$$

Note that it is not simply 2\*C columns! When C<=R, for instance, the user enters a **4** for C and **6** for R, your program should display a figure with **C** rows and **(2\*C-1)**-columns of "$" characters:

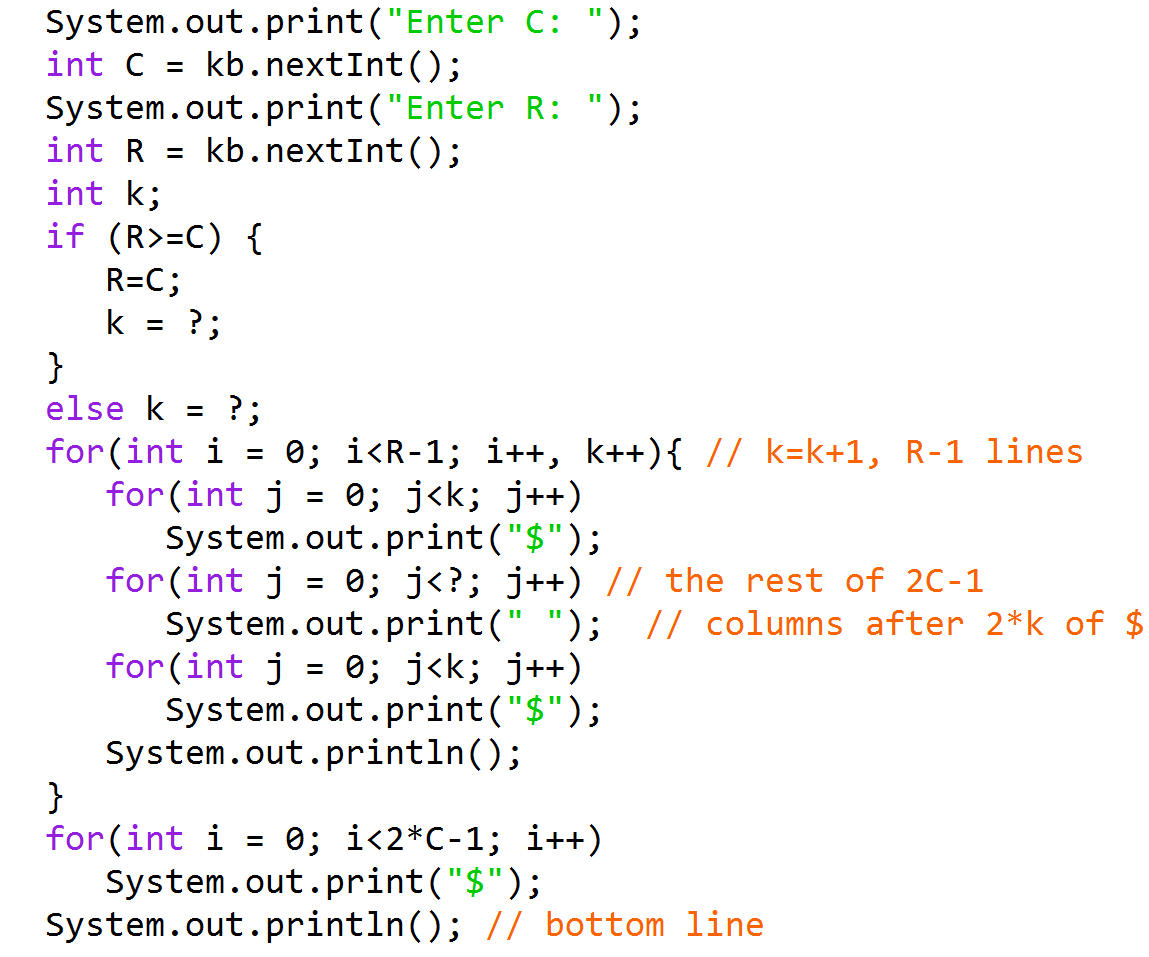
$ $

$$ $$

$$$ $$$

$$$$$$$

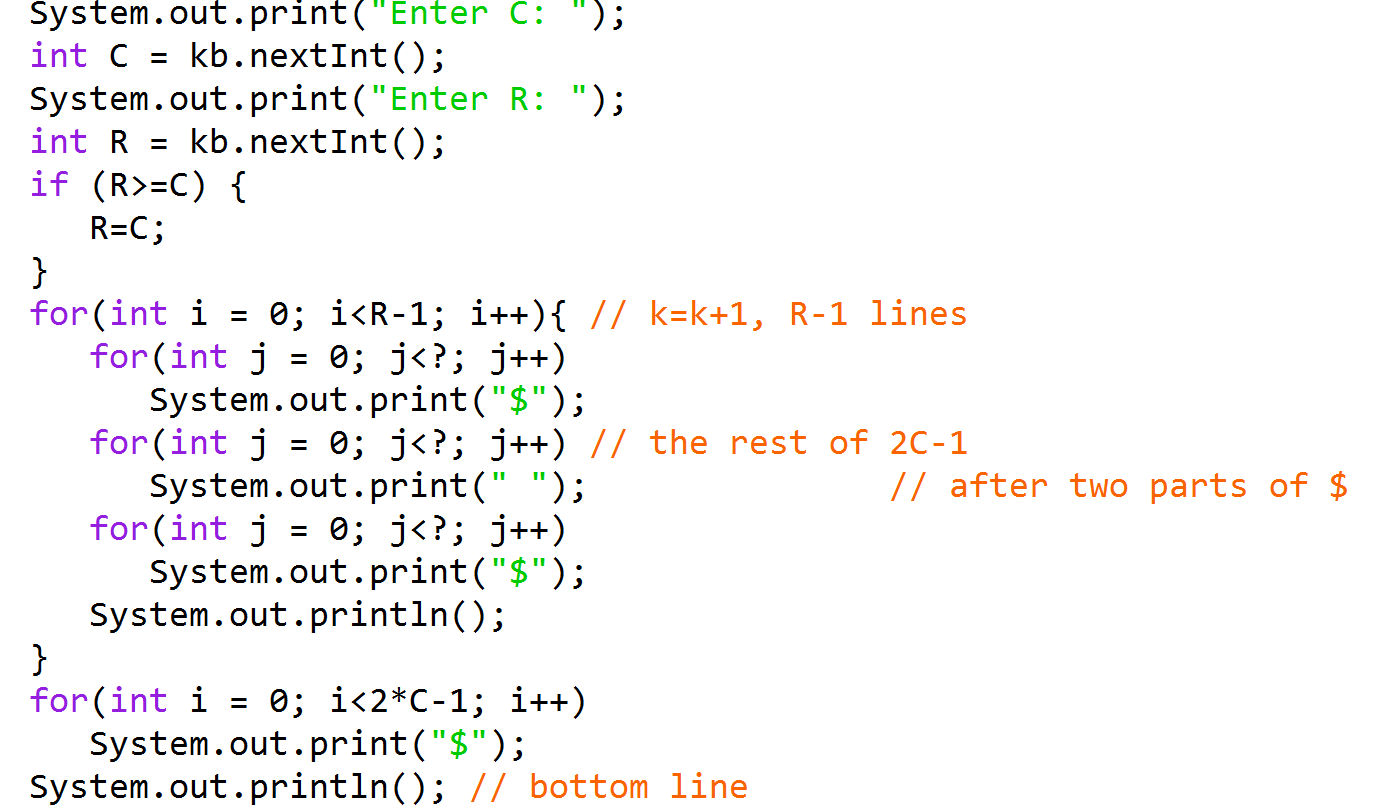
Step 1) Adopt the pattern in the picture:



Step 2) Each line, there are two parts of k-“$”s in display. When the loop starts to display the first line in i-loop, what is the initialization of k, for both the cases of “R>=C” and “else”?

Step 3) For each line, besides these two parts of $-display, the rest will be used for the “ ”s. How can we express this calculation (of the rest) here?

Step 4) Compare to the implementation with counter control loop:



Which version is easier to implement, steps1-3) or step 4? Submit your best version of code.