ESC101 : Fundamental of Computing

Lab Test for 13th November 2008

Instructions:

1. The duration of the test is 3 hrs (from **2:00 pm to 5:00 pm**).
2. **Directory Structure:** Create a directory in home of localusr, and name it with your roll number. e.g. If your roll number is $Y8001$, the directory should be named $Y8001$ (`Y` should be upper case). Inside this directory, create three directories named **One**, **Two** and **Three** corresponding to the programs for the three problems of this test. In directory **One**, create file $PowerfulNumbers.java$ for the java program for first problem. In directory **Two**, create files $LineSegment.java$ and $LineSegmentExample.java$ for the second problem. In directory **Three**, create file $Tile.java$ for the java program of the third problem.
3. Use of good coding practices (indentation, use of methods, and proper naming of variables and methods) carries weightage.

Problems:

1. **Powerful Numbers :** (marks = 12)
   A positive number n is said to be a *powerful number* if for every prime number p dividing n, p^2 also divides n. That is, if $p_1, ..., p_k$ are distinct prime factors of n such that $n = p_1^{a_1} p_2^{a_2} \dots p_k^{a_k}$, then n is powerful if $min(a_1, a_2, \dots, a_k) \geq 2$.

   For example, $1,\ 2^2,\ 3^3,\ 2^2 \cdot 3^2 \cdot 5^3$ are powerful numbers. But $2^2 \cdot 5$ is not a powerful number since the power of 5 is 1.

   Write a JAVA program that takes a positive integer n from command-line and prints whether it is a powerful number or not.

2. **Line Segment :** (marks = 12)
   A 2-dimensional line segment is defined by the coordinates of its endpoints. You have to design a JAVA class *LineSegment* that represents a 2-dimensional line segment. You have to use four attributes $x1, y1, x2, y2$ of type double for this class such that the pair $(x1, y1)$ corresponds to one endpoint, and the pair $(x2, y2)$ corresponds to another endpoint of the line segment. Write two constructors for this class.

   - One constructor is parameter-less (it does not take any argument), and it initializes the newly created line segment such that its one end point is $(0, 0)$ and another endpoint is $(1, 1)$.
   - The second constructor takes four arguments of type *double* and initializes the endpoints of the newly created line segment accordingly.

   You should design and implement the following four *non-static* methods:

   (a) **print**: Prints *this* line segment as: $((x1, y1), (x2, y2))$ where $(x1, y1)$ and $(x2, y2)$ are endpoints of *this* segment

   (b) **isVertical**: Returns **true** if *this* line segment is vertical and *false* otherwise.

   (c) **length**: Returns the length of *this* line segment.

   (d) **intersects**: Takes an object of the class *LineSegment* as argument, say R, and returns **true** if *this* line segment intersects R.

   Now, write a program (in the file $LineSegmentExample.java$) that uses the class *LineSegment* that you have designed. The program takes four command line arguments of type double and creates corresponding line segment. Let us denote this line segment by L. The program then creates another line segment R using the parameter-less constructor. The program computes the following information and prints appropriate messages:

(a) Print the line L.

(b) Determine whether L is vertical.

(c) Determine length of L.

(d) Determine whether L intersects M.

3. **Tiling a rectangular area** (marks=16)

There are the following three types of tiles.

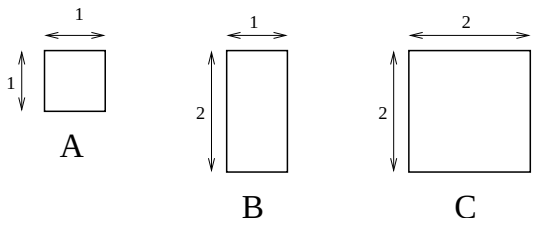

Figure 1: Three types of tiles

Given a $2 \times n$ rectangular area, we want to cover the area using the above types of tiles. The only restriction is that for each tile of type A being used in the area, there must be at least one tile of type A neighboring/adjacent to it, that is, they have one edge in common. You may assume that we have unlimited supply of tiles of each type. You have to write a program which reads n from command-line and computes the number of all possible ways to cover the $2 \times n$ area with these tiles. For making you more familiar with the problem, please study the two figures given below.

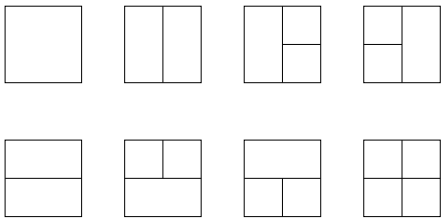

Figure 2: There are total 8 ways to tile a 2×2 area

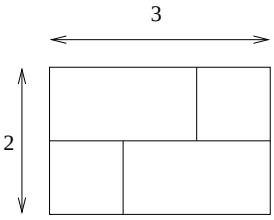

Figure 3: The covering given above is not a valid covering for a 2×3 area since the two tiles of type A used in it are not adjacent

Note : Proceed along similar lines as that of question 2 of quiz3. The solution of this quiz problem was posted on the course webpage, and was also explained in full detail during one of the optional extra classes.