ENGR 244 – ENGINEERING COMPUTING – SPRING 2018 – TERM EXAM II

Recommendation: Solve two out of the three given problems.

- You have **110 minutes** from the moment that the beginning of the exam is announced.
- This exam consists of three problems. Each is worth up to ten credits, out of 100 credits for the whole course. At most 20 credits can be gained from the present term exam.

  You need to **work on two problems** to achieve an optimal outcome.
- If you **choose to work on two problems**, these will both count normally (i.e., up to ten credits each), yielding an **optimum total of 20 credits**.
- If you choose to work on all three problems, the outcomes will be ordered by the number of credits achieved. **The best two problems count normally**, with up to ten credits each, and **the remaining problem does not count**, again yielding up to 20 credits for the exam as a whole. The outcome of the remaining problem does not influence the grade.
- Any concerns on scheduling, grading, or similar matters involving the ENGR department need to be addressed to the ENGR vice head of department R. Taha.

Make sure that **every paper** that you submit contains your **name** and **student ID**. Any access to means of communication is a case of cheating irrespective of what is communicated. Please **turn off your cell phones**. *You do not need to place your cell phones on the front desk.*

Recall that it is sufficient to **solve two out of the three present exam problems**. Feel free to hand in your submission at any time; leave the room without disturbing other participants.

AUIS student ID number: ____________________

Full name: __________________________________

American University of Iraq, Sulaimani

TERM EXAM II – PROBLEM 1

The following Java program determines, among geographic points on a *n* x *n* grid, where *n* is an integer given by the user, the grid point with the coordinates closest to the AUIS campus:

```
import java.util.Scanner;

public class ClosestGridPoint
{
  public static double[ ] AUIS_coordinates = {45.35, 35.57};  // 45.35° East, 35.57° North

  public static double objectiveFunction(double[ ] x)
  {
    double dx0 = x[0] - AUIS_coordinates[0];  double dx1 = x[1] - AUIS_coordinates[1];
    return dx0*dx0 + dx1*dx1;
  }

  public static void main(String[ ] args)
  {
    double[ ] x_min = {41, 29};  double[ ] x_max = {59, 38};

    // USER INPUT: grid size, in number of points per axis
    //
    System.out.print("Grid points on each axis: ");
    Scanner stdin = new Scanner(System.in);
    int n = stdin.nextInt( );  // get the user input for the variable n
    stdin.close( );

    // calculate distance between grid points
    //
    double[ ] step_size = new double[2];
    step_size[0] = (x_max[0] - x_min[0]) / (n - 1.0);
    step_size[1] = (x_max[1] - x_min[1]) / (n - 1.0);

    // determine the closest grid point
    //
    double g_longitude = 0.0;  double g_latitude = 0.0;
    double min_value = 1000000;  // initialize to a very large value
    double[ ] x = new double[2];
    for(int i = 0; i < n; i++)
    {
      x[0] = x_min[0] + i * step_size[0];  // longitude (grid)
      for(int j = 0; j < n; j++)
      {
        x[1] = x_min[1] + j * step_size[1];  // latitude (grid)
        double r2 = objectiveFunction(x);
        if(r2 < min_value)
        {
          min_value = r2;
          g_longitude = x[0];  g_latitude = x[1];
        }
      }
    }

    // PROGRAM OUTPUT: closest grid point coordinates
    //
    System.out.println( g_longitude  +  " East, "  +  g_latitude  +  " North" );
  }
}
```

a) How do the time requirements of this program scale asymptotically as a function of *n*, where *n* is the value specified by the user?

b) If the user input is 4, and hence, *n* is assigned the value 4, what is the program output?

TERM EXAM II – PROBLEM 2

Consider the following Java program:

```
import java.util.Scanner;

public class SimplifyThisCode
{
  // in part c) of this problem, this method needs to be replaced with a simpler
  // version; only the body of the method should be simplified; do not modify
  // the main method, do not modify the head of the method - only the body.
  //
  public static boolean simplifyThisMethod(int k)
  {
    if(0 >= k) return false;
    if(k == 1) return true;
    if(k == 2) return true;

    return simplifyThisMethod(k - 3);  // recursion, i.e., call to this method itself
  }

  public static void main(String[ ] args)
  {
    Scanner stdin = new Scanner(System.in);
    int n = stdin.nextInt( );  // take value for n from user input
    stdin.close( );

    if( simplifyThisMethod(n) ) System.out.println("True.");
    else System.out.println("False.");
  }
}
```

a) What methods does this code contain, and what is their return type?

b) Assume that the user input (value given to *n*) is 7. What is the program output?

c) Write a new code for the body of the method “simplifyThisMethod”, which is simpler than the present one, but has exactly the same outcome for all possible values of *n*.

The new code for the method “simplifyThisMethod” should not be recursive, i.e., it should not include a call to the same method itself; it should consist of three lines of code or less (body only, not including the head and the curly braces); and it should use the modulo operator (“%”) to determine the remainder after division where it is appropriate to do so.

TERM EXAM II – PROBLEM 3

Consider the following Java program:

```
import java.util.Scanner;

public class IntegerArithmetics
{
  public static void main(String[ ] args)
  {
    Scanner stdin = new Scanner(System.in);
    int n = stdin.nextInt( );  // take n from user input
    stdin.close( );

    int y = 0;
    for(int i = 1; n >= i; i++)  y = y + i;

    int x = 2*y - n*n;
    System.out.println( x );  // give x as program output
  }
}
```

a) How many local variables are declared in the main method, and what is their type? *Here, do not count the argument of the main method as a local variable.*

b) How do the time requirements of this program scale asymptotically as a function of *n*?

c) How do the space requirements (i.e., memory requirements) scale as a function of *n*?

d) Determine the loop invariant and the loop postcondition of the “for” loop.

e) Assume that the user input value, assigned to *n*, is 244. What is the program output?

Recall that $1 + 2 + ... + k = k(k + 1) / 2$.

ENGR 244 – ENGINEERING COMPUTING – SPRING 2018 – TERM EXAM II

AUIS student ID number: ______________________

Full name: ____________________________________

American University of Iraq, Sulaimani