**Lab 2 – Problem Solving and Python Coding**

**(70 Points)**

In this lab, we will write simple Python programs for two of the problems that were discussed in the lesson slides. We will also develop a solution for another problem using the problem analysis tools.

**Problem 1: (30 points)**

You are given the salaries of the 50 employees in a company, and you are required to compute the average salary value. Use the problem analysis tools to analyze this problem and develop a solution for it.

1. Draw a PAC chart that shows the given information, the required information, describes the processing that needs to be done, and the solution alternatives, if you can think of any.
2. Draw an interactivity chart that shows the different solution modules.
3. Draw an IPO chart for that problem to show the input (given information), processing, module reference, and the output (required information).
4. Write an algorithm for each module that you specify in the interactivity chart. The algorithm for your calculations part should contain the equation that you will use to compute the average value.

**Problem 2: (20 points)**

You are given three numbers, and you want to find the largest out of the three numbers. Use the problem analysis tools to analyze this problem and then write an algorithm for each module.

**Algorithms for Problem 2’s Modules:**

the Read Module:

1. Read number1
2. Read number2
3. Read number3
4. Initialize largest = -∞

the Compare Module:

1. If number1>number2 and number1>number3 then set largest = number1
2. Else, if number2>number1 and number2>number3 then set largest = number2
3. Else, set largest = number3

the Print Module:

1. Print the largest number

We will convert each step in each algorithm into a computer instruction that is written in Python language as follows.

* First, open IDLE. In the top menu, select File -> New
* Save the new file with any name.
* Take the code that is written in bold font below under each step, and type it in the new file.
* Save the new file.
* Run your code by pressing F5, or by going to the above menu and selecting Run -> Run Module.
* When your program runs, it will take you to the main Python console.
* On the python console, you need to input three values for the number1, number2 and number3. Just enter one value and press enter, three times.
* The largest out of the three values will be printed.

the Read Module:

1. Read number1  
   We use the input() function to read an input from the user, and use the int() function to convert that user input into an integer.   
   **number1 = int( input("Enter a number:") )**
2. Read number2  
   **number2 = int( input("Enter a number:") )**
3. Read number3  
   **number3 = int( input("Enter a number:") )**
4. Initialize largest = -∞  
   We will set largest to a very small value, say, -10^10.  
   **largest = -1e10**

the Compare Module:

1. If number1>number2 and number1>number3 then set largest = number1  
   **if number1 > number2 and number1 > number3:  
    largest = number1**
2. Else, if number2>number1 and number2>number3 then set largest = number2  
   **elif number2 > number1 and number2 > number3:  
    largest = number2**
3. Else, set largest = number3  
   **else:  
    largest = number3**

the Print Module:

1. Print the largest number  
   **print(largest)**

**Problem 3: (20 Points)**

You are assigned the task of ordering paper boxes and printing copies of the board meeting report given the number of pages in the report, the number of meeting attendees, and the number of papers per paper box. You can only order whole boxes. You need to print five extra copies.

Do the same steps as in problem 2, by typing each python instruction in a new file in IDLE, and running your code. Notice how each step in the algorithm is written in python code.

**Algorithms (and python code) for the read, compute and print modules:**

1. First, you need to import the math library, which contains the rounding functions. You need to type:  
   **import math**
2. Set number\_reports  
   **n\_reps = 100**
3. Set number\_extra\_reports  
   **n\_extra\_reps = 25**
4. Set number\_pages  
   **n\_pages = 20**
5. Set the number of papers per box  
   **n\_papers\_box = 200**
6. Set number of required papers = (number of reports + number of extra reports) \* number of pages per report.  
   **n\_req\_papers = (n\_reps + n\_extra\_reps) \* n\_pages**
7. Set number of required boxes = (number of required papers / number of papers per box) rounded up to the nearest integer.   
   **n\_req\_boxes = math.ceil(n\_req\_papers / n\_papers\_box)**
8. Print number of required boxes.  
   **print(“The number of required paper boxes = ”, n\_req\_boxes)**

**Lab Deliverable:**

* For problem 1, include all your charts and algorithms.
* For problems 2 & 3, take screen shots of your code and the obtained output after you run it.
* Put everything in one doc/docx file and submit it.