National University of Computer and Emerging Sciences, Lahore Campus

Course:	**PF (Lab)**	**Course Code:**	**CL 118**
Program:	**BS (Computer Science)**	**Semester:**	**Fall 2019**
Duration:	**2 Hours**	**Total Marks:**	**60 (20+20+20)**
Paper Date:	**17-Oct-2019**	**Weight**	**30%**
Section:	**C,D,K,L,E, F**	**Page(s):**	**2**
Exam:	**Lab Midterm Exam**	**Reg. No**	

Important Instructions (Please read them before attempting the exam):

- Submit **ONLY .cpp File** in this format (Make the File **named** with your **Roll Number** e.g., L19-4152).
- **Plagiarism** will result in **F grade** in lab.
- No cell phones are allowed. Sharing of **USBs** or any other items is **not allowed.**
- Submission path will be announced soon.
- Submit your files on **\\cactus\Xeon\Fall 2019\Shakeel Zafar\PF MID Exam**
- Use **Visual Studio 2012.**

Question # 1:

You are asked to write a c++ program in which a user wants to know the k^{th} largest element. For that user initially enters the value of K. Then the user enters at least K+1 unique values in sorted order (remember sorted order can be both ascending or descending). The restriction in program is, you cannot use Array. You will find the kth largest if and only if the values were in sorted order. If the entered values are not in sorted order then just print that "the values or not in sorted order". If the values were in sorted order you have to tell the kth largest element.

Question # 2:

Write a program to calculate the place value of digit in an integer. For example, if the user inputs an integer 56918 and you want to determine the place value of 6, the output would be "Thousands". You can include a check for whether that specific digit is present or not. Your program should work for a maximum place value of "ten million".

Possible place values are *Units, tens, hundreds, thousands, ten thousand, hundred thousand, millions, ten million.*

Question # 3:

A high school has 1000 students and 1000 lockers, one locker for each student. On the first day of school, the principal plays the following game: She asks the first student to go and open all the lockers. She then asks the second student to go and close all the even-numbered lockers. The third student is asked to check every third locker. If it is open, the student closes it; if it is closed, the student opens it. The fourth student is asked to check every fourth locker. If it is open, the student closes it; if it is closed, the student opens it. The remaining students continue this game.

In general, the nth student checks every nth locker. If the locker is open, the student closes it; if it is closed, the student opens it. After all the students have taken their turn, some of the lockers are open and some are closed. Write a program that prompts the user to enter the number of lockers in a school. After the game is over, the program outputs the number of lockers that are opened. Test run your program for the following inputs: 1000, 5000, 10000. Do you see any pattern developing?
(**Hint:** Consider **locker** number 100. This locker is visited by **student numbers** 1, 2, 4, 5, 10, 20, 25, 50, and 100. These are the positive divisors of 100. Similarly, locker number 30 is visited by student numbers 1, 2, 3, 5, 6, 10, 15, and 30. Notice that if the **number** of **positive divisors** of a locker number is **odd**, then at the end of

the game, the locker is **opened**. If the **number** of **positive divisors** of a locker number is **even**, then at the end of the game, the locker is closed.)