# Data Structures and Algorithm
# Lab 1

**Marking:**

- Remember to follow the programming style guidelines (e.g. All input/output file names should be defined as constants).
- You need to show to you tutor all of the programs and demonstrate that you understand them. The latest deadline is the following week lab, after that no mark will be given.
- Zip all of your programs to a file YourStudentID_YourName.zip and submit it using the subject website at bbc.vnu.edu.vn.

**Q1.** Using INPUT/OUTPUT streams to do the following tasks:

1. Write a program named *GenNum.cpp* to generate a sequence of n integers where n is also a randomly generated integer. Write the generated sequence to a file DAYSO1.TXT in the following format:
    - The first line contains the number n.
    - The following n lines contain n numbers in the sequence where each number is on a separate line.
2. Write a program named *ReadNum.cpp* to read a sequence of numbers from file DAYSO1.TXT with the following format:
   + The first line contains number n – the sequence's length.
   + The following n lines contain n integers where each integer is on a separate line.

   Your program should also perform the following tasks:
   a) Print the largest integer in the sequence.
   b) Print the second largest integer in the sequence.
   c) Print the $k^{th}$ largest integer in the sequence where k is read from the standard input.

**Q2.** A given file DAYSO2.TXT contains integers where each integer is on a separate line (the number of integers is smaller than 10000).

1. Write a program named Reverse.cpp to read all integers from file DAYSO2.TXT into an array and write all integers to a file KETQUA1.TXT in reverse order where each integer is on a separate line.
2. Let n be the number of integers in DAYSO2.TXT. Write a program called CutK.cpp that reads all integers from DAYSO2.txt in sequence and takes a value k (0<k<=n) from standard input to write the sequence to a file KETQUA2.TXT in the following order where each integer is on a separate line:
    - $(k+1)^{th}$ … $n^{th}$ numbers.
    - $1^{st}$, $2^{nd}$, … $k^{th}$ numbers.

**Q3.** Write a program named Complex.cpp to define the Complex class and its operators as described in the tutorial (Q3).