Data Structures & Algorithms *COURSE* & *LAB* – Spring 2022

(BS-IT-F20 Morning & Afternoon)

Assignment # 2

*Assigned on: **Friday, April 22, 2022***

*Submission Deadline: **Tuesday, April 26, 2022 (till 11:00 PM)***

Instructions:

- **This is an individual assignment. Absolutely NO collaboration is allowed. Any traces of plagiarism/cheating would result in an "F" grade in this course.**
- Do **NOT** copy even a single line of code from any other person or book or Internet or any other source.
- This assignment needs to be submitted in **Soft** form. The **Submission Procedure** is given below.
- Late submissions will NOT be accepted, whatever your excuse may be.
- This Assignment will be counted towards your *Sessional marks* in both **DSA-Course** and **DSA-Lab**.

Submission Procedure:

i. Create an empty folder. Put all of the ***.CPP*, .H, and input file(s)** of your assignment in this folder. Do NOT include any other files in your submission, otherwise your submission will NOT be graded. Don't forget to **mention your Name, Roll Number and Section in comments at the top of each CPP file**.

ii. Now, compress the folder (that you created in previous step) in **.RAR** or **.ZIP** format. The name of the RAR or ZIP file should be ***exactly*** according to the format:

`Mor-A2-BITF20M123` (for Morning section) OR

`Aft-A2-BITF20A456` (for Afternoon section),

where the text shown in **BLUE** should be your **complete Roll Number**.

iii. Finally, submit the (single) **.RAR** or **.ZIP** file through **Google Classroom**.

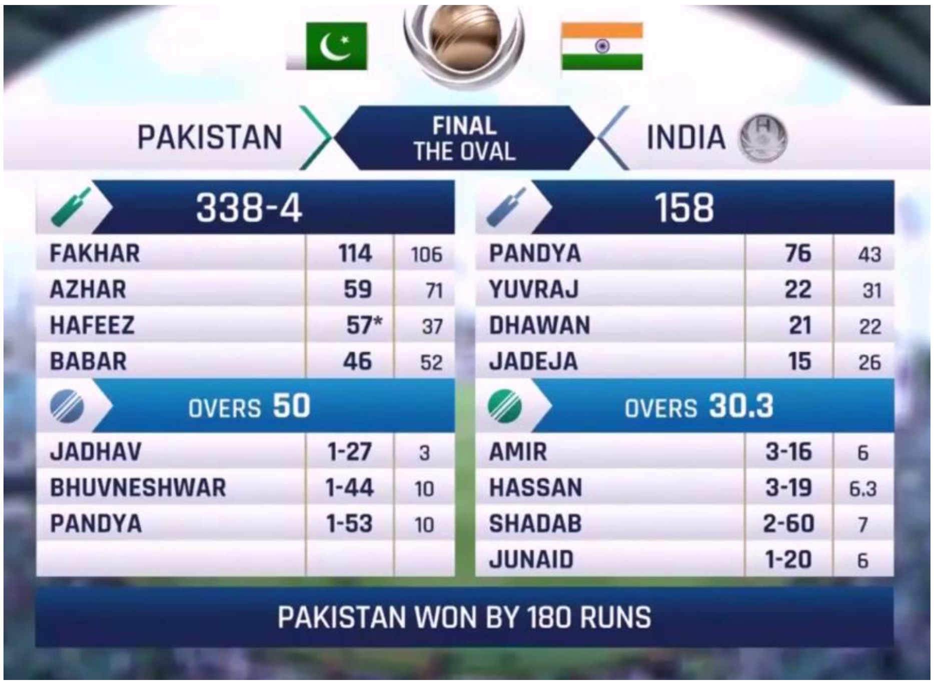
Note: *If you do not follow the above submission procedure, your Assignment will NOT be graded and you will be awarded ZERO marks.*

These *good programming practices* will also have their (significant) weightage in the marking of this assignment:

- Comment your code intelligently. **Uncommented code will NOT be given any credit.**
- Indent your code properly.
- Use meaningful variable and function names. Use the **camelCase** notation.
- Use meaningful prompt lines/labels for input/output.

If your program **does NOT compile correctly** or it **generates any kind of run-time error** (dangling pointer, memory leak etc.) you will get **ZERO marks** in the whole assignment.

In the match summary, which is displayed at the end of a cricket match, bowling figures of bowlers are displayed in **decreasing order of Wickets** that they have taken. Furthermore, if two (or more) bowlers have taken the same number of wickets, then the bowler conceding **lesser runs** is displayed first. As an example, see the following match summary of the final match of the ICC Champions Trophy 2017 (played at The Oval, London, on June 18, 2017):



PAKISTAN — FINAL THE OVAL — INDIA

338-4			158		
FAKHAR	114	106	PANDYA	76	43
AZHAR	59	71	YUVRAJ	22	31
HAFEEZ	57*	37	DHAWAN	21	22
BABAR	46	52	JADEJA	15	26
OVERS 50			OVERS 30.3		
JADHAV	1-27	3	AMIR	3-16	6
BHUVNESHWAR	1-44	10	HASSAN	3-19	6.3
PANDYA	1-53	10	SHADAB	2-60	7
			JUNAID	1-20	6

PAKISTAN WON BY 180 RUNS

You can see that the bowling figures of bowlers are displayed in **decreasing order of wickets** taken. In the second innings, two Pakistani bowlers have taken 3 wickets each, so their bowling figures are displayed in **increasing order of runs** that they have conceded.

In this assignment, you are required to sort a list (array) of bowling figures using various sorting algorithms. You must use the following struct `BowlingFigures` in your implementation:

```
const int NAME_LENGTH = 25;

struct BowlingFigures {
      char name[NAME_LENGTH+1];     // Name of the bowler
      int wickets;                  // Wickets taken by the bowler
      int runs;                     // Runs conceded by the bowler
};
```

Your program should take its input from a text file. A sample input file[1] is shown below:

```
6
Aamer Sohail
0 49
Aaqib Javed
2 27
Wasim Akram
3 49
Ijaz Ahmed
```

[1] In case you are wondering, these are the bowling figures of Pakistan's bowlers from the Final of the Cricket World Cup 1992 (which was played at MCG, Melbourne, Australia on March 25, 1992)

```
0 13
Mushtaq Ahmed
3 41
Imran Khan
1 43
```

The first line in the input file contains a positive integer indicating the number of bowling figures present in the file. Each bowler's bowling figure is on two lines in the input file. The first line contains the name of the bowler, while there are two integers on the second line which indicate the number of wickets taken and the runs conceded by the bowler, respectively. You can assume that the name of a player will contain a maximum of 25 characters.

Your program must have the following functions:

- **`BowlingFigures* readFromFile (char* fileName, int& count);`**
  This function will take the name of an input file and a reference (alias) of un-initialized integer variable as parameters. If the input file is not found/opened, this function should return NULL. If the file is opened successfully, this function should dynamically allocate an array of **`BowlingFigures`** and read all the bowling figures into this array. At the end, this dynamically allocated array of BowlingFigures will be returned from this function. This function should also store the count of the No. of bowling figures into the reference parameter (**`count`**).

- **`void printBowlingFigures (const BowlingFigures * bf, int count);`**
  This function will take the array of **`BowlingFigures`** and its size as parameters, and it will display all the bowling figures on the screen, in a neat and readable way.

- **`void sortBowlingFigures1 (BowlingFigures * bf, int count);`**
  This function will take the array of **`BowlingFigures`** and its size as parameters, and it will sort this array into **decreasing order** of bowling figures i.e. the bowling figures must be arranged in **decreasing order of wickets** taken, and if two (or more) bowlers have taken the same number of wickets, then the bowling figures of such players must be arranged in **increasing order of runs** conceded. You MUST use **Insertion Sort** to sort the array of bowling figures. You can assume that in the input file, no two bowling figures will be exactly identical.

- **`void sortBowlingFigures2 (BowlingFigures * bf, int count);`**
  This function will also sort the array of **`BowlingFigures`**, as explained in the description of previous function. However, this function will use **Selection Sort** to sort the array of BowlingFigures.

- **`void sortByName (BowlingFigures * bf, int count);`**
  This function will sort the array of bowling figures in increasing order w.r.t. **the names of the bowlers**. In this function, you must use the **Bubble Sort** algorithm to sort the array of BowlingFigures.

For the input file shown on the previous page, a sample run of your program might produce the following output:

```
Enter the name of input file: cricket.txt

Following 6 Bowling figures were read from the input file:

       Aamer Sohail  0-49
        Aaqib Javed  2-27
        Wasim Akram  3-49
         Ijaz Ahmed  0-13
      Mushtaq Ahmed  3-41
         Imran Khan  1-43

               --------
                 Menu
               --------
1. Sort BowlingFigures using Insertion sort
2. Sort BowlingFigures using Selection sort
3. Sort BowlingFigures by Name (using Bubble sort)
4. Quit

Enter your choice: 3

The 6 Bowling figures after sorting by NAME are:

       Aamer Sohail  0-49
        Aaqib Javed  2-27
         Ijaz Ahmed  0-13
         Imran Khan  1-43
      Mushtaq Ahmed  3-41
        Wasim Akram  3-49


Enter your choice: 1

The 6 Bowling figures after sorting using Insertion sort are:

      Mushtaq Ahmed  3-41
        Wasim Akram  3-49
        Aaqib Javed  2-27
         Imran Khan  1-43
         Ijaz Ahmed  0-13
       Aamer Sohail  0-49

Enter your choice: 4
```

☺ GOOD LUCK! ☺