SCHOOL OF ARCHITECTURE, COMPUTING & ENGINEERING

Submission instructions

- Cover sheet to be attached to the front of the assignment when submitted
- Question paper to be attached to assignment when submitted
- All pages to be numbered sequentially
- All work has to be presented in a ready to submit state upon arrival at the ACE Helpdesk. Assignment cover sheets or stationery will **NOT** be provided by Helpdesk staff

Module code	**CN5121**		
Module title	**Data Structures and Algorithms**		
Module leader	**Paolo Falcarin**		
Assignment tutor	**Rawad Hammad, Amin Karami**		
Assignment title	**Coursework**		
Assignment number	**1**		
Weighting	**50%**		
Handout date	**31 / 10 / 2019**		
Submission date	**18 / 12 / 2019**		
Learning outcomes assessed by this assignment			
Turnitin submission requirement	**YES**	**Turnitin GradeMark feedback used?**	**NO**
UEL Plus Grade Book submission used?	**NO**	**UEL Plus Grade Book feedback used?**	**NO**
Other electronic system used?	**no**	**Are submissions / feedback totally electronic?**	**yes**
Additional information	ASSESSMENT FEEDBACK - Feedback on your assessment will be available in four working weeks from the submission date. Please refer to the module pages on Moodle for assessment specific details.		

Form of assessment:

□ Individual work ☒ Group work

For **group work** assessment which requires members to submit both individual and group work aspects for the assignment, the work should be submitted as:

☒ Consolidated single document □ Separately by each member

Number of assignment copies required:

☒ 1 □ 2 □ Other

Assignment to be presented in the following format:

☒ On-line submission
□ Stapled once in the top left-hand corner
□ Glue bound
□ Spiral bound
□ Placed in a A4 ring bound folder (not lever arch)

Note: To students submitting work on A3/A2 boards, work has to be contained in suitable protective case to ensure any damage to work is avoided.

Soft copy:

□ CD (to be attached to the work in an envelope or purpose made wallet adhered to the rear)
□ USB (to be attached to the work in an envelope or purpose made wallet adhered to the rear)
☒ Soft copy not required

Note to all students

Assignment cover sheets can be downloaded from logging into UEL Direct via the following pathway.

UEL Direct → **My Record** → **My Programme** → **Assessment log dates with Barcoded Frontsheet**

All work has to be presented in a ready to submit state upon arrival at the ACE Helpdesk. Assignment cover sheets or stationery (including staplers) will NOT be provided by Helpdesk staff. This will mean students will not be able to staple cover sheets at the Helpdesk.

CN5121 Data Structures and Algorithms
Assignment 2019/20 Academic Year

This group coursework carries 50% of the overall module marks and it is composed of two components

1. **group-based programming task (60%)**
2. **Weekly assessment tasks (40%)**

Each group should have at most **4 members**.
This coursework was designed keeping this group size in mind and the size will be enforced unless there are exceptional circumstances.
You will also be required to designate a ***group leader***. It is vital that each group nominates a group leader as early as possible because the group leader will be responsible for all group communication with the module leader and for the final submission of the group work to Moodle.
Please note that group communication should only be done through the group leader unless there are exceptional circumstances.
The deadline for finalising your group is **8/11/2019**.
Each group leader must send an email to the module leader on or before this date confirming details of the members of the group. Please include ***Student ID*** and ***Full Name*** for each member of the group and clearly identify the group leader.
If you are not in a group by the **deadline** you will be randomly assigned to one.
If your name is in multiple groups, you will also be randomly assigned to one.
In former iterations of this module, it has been observed that students that form their own group tend to do better, therefore, you are strongly advised to form your own groups.
A final list of groups along with details each member will be published at the Moodle module site; this will be the **final** official allocation and will be strictly followed.

1. Group-based problem (60 marks)

You are required to develop a Java application to find the optimal solution for the Research Evaluation Framework (REF).

Each university in UK is required every 7 years to submit their best research papers to the REF reviewers and they will get a score called GPA (Grade Point Average) that will affect the research funding and university league table for the following years.

The REF independently assesses different research units within the same institution: for example, unit 11 is Computer Science, unit 15 is Engineering, unit 16 is architecture, and so on. Each research paper can be written by different co-authors (usually from 1 to 7 co-authors). Each unit (or department) employs a certain number of staff who are employed full time or part-time: one full-time equivalent (FTE) staff is a lecturer working full-time, while, for example, part-time lecturers working one day a week are 0.2 FTE (20%).

Each unit is required to submit a total of research papers 2.5 times the total FTE: thus, in case of 10 FTE, the unit will submit 25 papers.

Authors constraints: each staff member of a unit can submit from 1 to 5 papers as main author, but they can appear as a co-author in papers where other colleagues are designated as the main author. You can include papers from authors who left the institution: they are not subject to the constraint of 1 to 5 papers per staff, and they do not increase the total head-count of FTEs (i.e. the overall number of authors).

Each paper has been assessed at least once in different annual reviews and they got a score from 0 to 12 from different reviewers; thus, one paper might have got two different scores from 2 different reviewers, and you might want to consider the average score.

A research paper is published in an international journal or in a conference proceedings book; journals and conferences are also ranked in terms or their quality using metrics like impact factor and they usually belong to 4 different quartiles: first quartile contains the best and more cited journals, while the 4th quartile contains the less cited journals.

Create a java program able to implement the following specification:

a) Read a text file to load the data into a data structure in memory. **(15 marks)**

b) Given all the authors, print on file the related best N papers according to their maximum average review score: in case of same score between 2 papers, choose the ones published in a journal/conference in the best quartile. **(10 marks)**

c) Find the best N papers considering the above-mentioned authorship constraints: at least one paper per author, at most 5 papers (**5 marks).**

d) Calculate the Grade-Point Average (GPA) of the N chosen papers **(5 marks)**;
The GPA is the weighted average of the chosen N papers weighted by their REF score (integer number 0 to 4). In the REF assessment, papers are scored from 0 to 4 stars, while the reviewers scored from 0 to 12: thus, the following mapping can be assumed:

Reviewer	REF-score
0	0
1	1 Lower
2	1 medium
3	1 Upper
4	2 Lower
5	2 Medium
6	2 Upper
7	3 Lower
8	3 Medium
9	3 Upper
10	4 Lower
11	4 Medium
12	4 Upper

e) Given a number M of authors (less than the total), find the best set of N (= 2.5*M) papers to be submitted, the ones that maximize the GPA, and visualize the corresponding subset of authors. **(10 marks)**

f) Save the N best papers found in a csv output file with the format **(5 marks)**

g) Design a text-based menu or a Graphical User Interface to let the user choose the various options above and visualize the results **(10 marks)**

Input file

The first line contains the list of the authors to be considered for the submission

Then the following lines contain information about the papers, more precisely the information of one paper covers three lines.

The first line contains the paper id, the title of the paper, and the quartile of the journal in which it is published, separated by commas.The second line contains the staff authors list in comma separated list, while the third line contains the scores of the annual reviews, at least one number.

In the example below, the first line states that there are M=3 authors “AK”, “PF”, and “AM”. The text from the second line to the fourth line contain data about Paper1, then paper2, and so on.

Each paper has an id (for example “Paper1”), a title (“Title1”) and the quartile of the journal/conference in which it has been published (“1”); then the next line contains the co-authors of that paper (AM and AP); the last line contains the scores of the reviews, in this case three reviews.

The following three lines refer to Paper2, authored by AM, PS, and TC which got only 2 reviews

Please not that paper4 has one author “JD” which does not belong to the initial authors list at the beginning of the file: this represents a paper from an author who left the institution and then their papers can be added to the submission but they are not subject to the constraint of 1 to 5 papers per staff, and they do not increase the total head-count of FTEs.

A bigger input file will be provided later on.

```
AK, PF, AM
Paper1, Title1, 1
AM, AP
9, 8, 7
Paper2, Title2, 2
AM, PS, TC
10, 8
Paper3, Title3, 3
AM, PP
1, 2, 4
Paper4, Title4, 2
JD
6
```

Output file

The selected N best papers should be printed on a csv text file containing one line per paper in descending order of score, after the first line containing the GPA. The example below in case of N=3.

```
GPA = 2.66
Paper2, Title2, 9
Paper1, Title1, 8
Paper4, Title4, 6
```

Please notice that 9 and 8 average review scores are mapped to 3 star REF score while 6 is mapped to 2; thus the GPA is (2*3 +1*2)/3 = 2.66 because I have two 3* papers and one 2* paper on a total of N=3 chosen papers at the denominator.

2. Weekly assessments (40 marks)

From week2 to week10 there will be a weekly individual assessment at the end of each practical lab session, consisting of a short Moodle quiz (15 minutes) or the submission of a piece of java code.

Each weekly assessment will be worth 5 marks, and the best 8 marks out of the total 10 assessments will be used to calculate the marks for this second component, in order to mitigate the effect of possible absences.

3. Presentation and Submission

Each group is required to **submit** one zip file including all the source code (and compiled code) in a zip file named with the group number, by the **18th of December**.

Presentations

The schedule of the presentations will start in the week of the submission deadline and the tentative date allocated for the demos is **Thursday 19th December**; the group will show their code running to the lecturers showing how each requirement has been implemented.

Then each student will answer few questions about their code and will clarify the distribution of work among the group; the lecturers might later run testcases on the submitted code to verify how many requirements have passed.

Note Well

- If the code does not compile, the task will be marked zero.

Marking scheme	marks
Problem 1 (demo)	**60**
Ten Weekly assignments	**40**

Plagiarism

The University defines an assessment offence as any action(s) or behaviour likely to confer an unfair advantage in assessment, whether by advantaging the alleged offender or disadvantaging (deliberately or unconsciously) another or others. A number of examples are set out in the Regulations and these include:

“D.5.7.1 (e) the submission of material (written, visual or oral), originally produced by another person or persons, without due acknowledgement, so that the work could be assumed the student’s own. For the purposes of these Regulations, this includes incorporation of significant extracts or elements taken from the work of (an) other(s), without acknowledgement or reference, and the submission of work produced in collaboration for an assignment based on the assessment of individual work. (Such offences are typically described as plagiarism and collusion.)”

The University's Assessment Offences Regulations can be found on our web site. Also, information about plagiarism can be found on the programme’s handbook.