

CENG205 DATA STRUCTURES

ASSIGNMENT 2

- 1) Sort the records (according to **course codes**) in the "Data.dat" file **ascendingly**, which contains the course name and course code information about Gazi University Computer Engineering Department courses (see Figure 1). A **linked list** must be used to store records in order in memory. A linked list data structure should be created and all records (except **duplicate** records) in the file should be added to this linked list sequentially. Duplicate records in the file should be added to the linked list only **once**. After the records in the file are read, they will be printed sequentially both on the screen and in the "Sorted.dat" file. While printing the sorted course data, course codes must be printed before course names with a **tab character** separating them (see Figure 2). There should be no duplicate records in the "Sorted.dat" file.

!!! Solutions made using array will not be evaluated!!!

```

BILGISAYAR PROGRAMLAMA I 101
DATA STRUCTURES 205
BILGISAYAR MUHENDISLIĞINE GIRIS 103
OBJECT ORIENTED PROGRAMMING 213
DATA STRUCTURES 205
GORUNTU ISLEMEYE GIRIS 471
YONEYLEM ARASTIRMASI 488
INTRODUCTION TO AI 455
INTRODUCTION TO ML 476
SYSTEM ENGINEERING 481
GEO. INFORMATION SYSTEMS 463
SCRIPTING LANGUAGES 367
ELEKTRİK VE ELEKTRONİK DEVRELER 211
INTRODUCTION TO DATA SCIENCE 313
İSLETİM SİSTEMLERİ 309
ALGORITHM ANALYSIS AND DESIGN 368
INTERNET PROGRAMLAMA 359
MUHENDİSLİK PROJESİ 315
GRAF TEORİSİ 358
BILGISAYAR MUHENDISLIĞINE GIRIS 103
JAVA PROGRAMLAMA 372
OLASILIK VE İSTATİSTİK 207
BILGISAYAR MİMARİSİ 311
INTRODUCTION TO ML 476
BILGISAYAR MUH. ÖZEL KONULAR I 459
DATA STRUCTURES 205
VERİ İLETİSİMİ 403
DATA MINING 489
TASARIM ORUNTULERİ 483
DISTRIBUTED SYSTEMS 465
INTERNET PROGRAMLAMA 359

```

Figure 1. "Data.dat" file contents

```

101    BILGISAYAR PROGRAMLAMA I
103    BILGISAYAR MUHENDISLIĞINE GIRIS
205    DATA STRUCTURES
207    OLASILIK VE İSTATİSTİK
211    ELEKTRİK VE ELEKTRONİK DEVRELER
213    OBJECT ORIENTED PROGRAMMING
309    İSLETİM SİSTEMLERİ
311    BILGISAYAR MİMARİSİ
313    INTRODUCTION TO DATA SCIENCE
315    MUHENDİSLİK PROJESİ
358    GRAF TEORİSİ
359    INTERNET PROGRAMLAMA
367    SCRIPTING LANGUAGES
368    ALGORITHM ANALYSIS AND DESIGN
372    JAVA PROGRAMLAMA
403    VERİ İLETİSİMİ
455    INTRODUCTION TO AI
459    BILGISAYAR MUH. ÖZEL KONULAR I
463    GEO. INFORMATION SYSTEMS
465    DISTRIBUTED SYSTEMS
471    GORUNTU ISLEMEYE GIRIS
476    INTRODUCTION TO ML
481    SYSTEM ENGINEERING
483    TASARIM ORUNTULERİ
488    YONEYLEM ARASTIRMASI
489    DATA MINING

```

Figure 2. The expected "Sorted.dat" file contents

Useful Information:

In order to process the files:

- Defining a file pointer: `FILE *file_pointer_name;`
Example: `FILE *fileptr;`
- Opening the file in the appropriate mode: `fopen ("name of the file" , "mode");`
`file_pointer_name = fopen ("name of the file" , "mode");`

Table 1. mode and meaning

Mode	Meaning
w	<ul style="list-style-type: none"> Opens the text file in write mode. If the file does not exist, it is created, if there is, the records in the file are deleted and opened again.
r	<ul style="list-style-type: none"> Opens the text file in read mode. The file must already exist.
a	<ul style="list-style-type: none"> The text opens the file in insert mode. Newly entered records are written at the end of the file. The file does not need to exist before.

Example: `FILE *fileptr;`
`fileptr = fopen ("Data.dat", "r");`

- Reading from or writing to file: `fprintf / fscanf (stream, "format_text", list of variables);`

Example: `fprintf (fileptr, "%d", course_code);`
`fscanf (fileptr, "%d", &course_code);`

- Closing the file: `fclose (file_pointer);`

Matters needing attention:

- **Zip** the assignment file. Zip File Naming: NameSurname_Studentnumber (E.g- SumeyraKus_141180048)

(Name and surname information must be included in the file. Necessary information should be written separately as a description line (comment line) in the code file submission.)

- Submissions should contain **only** the **code files** but **not** the **executables**.
- Assignments must be uploaded via **Gazi LMS**.
- Assignments should be prepared in **English**.
- Codes in assignments should be written in **C programming language**.
- Coding standards should be followed, and full attention should be paid to details such as indentation, variable/function naming. It should be detailed with comment lines.
- Assignments must be submitted **on time**.
- Low points are taken off for sloppy or copy-paste assignments.
- Must no copy, zero will be given when a copy is detected.

(Assignments will be subject to **pairwise code comparison** and similar assignments will be considered as copies.)