



**AYBU CENG114**  
**Take Home Exam #1**  
**(Midterm)**  
(100 points)

**Due: Friday, 10:00 pm April 24, 2020**

**General Information:** This exam is designed to show you the ability to use the topics we discuss in class and the concepts used in presentations, and to show your ability to write code to support your ideas. It is possible to interpret the questions in various ways; but first make sure you answer the question I gave you, and then definitely add a comment on your behalf and write your opinion for each question. Presenting your own opinion will be an indication that your homework is an original work.

Finally: This exam will replace midterm exam. You only need to upload and send the exam on the AYBUZEM system within the given time. Be careful not to leave it to the last minute. If you are having problems with AYBUZEM, contact me as soon as possible via e-mail address [evrenaykac@gmail.com](mailto:evrenaykac@gmail.com).

**Grading:** I will be grading these exams by looking at your answers: the clarity of your thoughts, the ability to use specific sections from sources to support your answers, your ability to clearly explain how your solution code relate to the question I ask. The correctness and error-free compilation of the codes you write will still be very important.

In addition, be sure to meet the file format requirements, or you may see a drop in your score. Each question has an equal weight in points. Format requirements are as follows:

- 1 inch margins on all sides
- 12 pt Times New Roman font
- For the code segments use **Courier New** font style

Let me know if you have any questions. Good luck.

*Questions can be answered using the resources of our course on the AYBUZEM website.*

Write programs to solve the following problems.

Choose and answer 13 of the 14 questions you want. Only the first 13 questions you answer will be scored.

- 1) Given the names of 10 persons as first name(s) followed by surname, display their names as surname followed by firstname(s). Notice that, some people may also have some middle names.
- 2) Given the names of 10 persons as first name(s) followed by surname, display their names by abbreviating any middle name(s). (For ex., if the name is "ALI CAMCI", it will not change, but if it is "ALI HASAN CAMCI", it will change as "ALI H. CAMCI")
- 3) Given a sentence, delete only the last "THE" from that sentence if it starts with the word "WE", otherwise delete all "THE"s.

4) Given a sentence, display

- the number of "ST"s
- the number of words starting with "ST" (such as START, STYLE, etc.)
- the number of words ending with "ST" (such as FIRST, BEST, etc.)

5) Given a sentence,

- Delete all 3-letter words.
- Move the first word to the end of the sentence.
- Reverse each word.
- Exchange the first and last words.

Display the sentence after each step.

6) ABC Bank account numbers consist of a four-digit branch code representing the branch in which that account is opened, followed by a seven-digit number (e.g. 42760054418). Given the codes and the names of the Ankara branches of ABC Bank in file "Branches.txt", and 100 account numbers and the money in each account in file "Accounts.txt", and the name of a branch as input, display how many of those accounts are opened in that Branch, and the sum of the money in those accounts. (Branch: Şube)

7) XYZ Bank account numbers consist of a four-digit branch code representing the branch in which that account is opened, followed by a seven-digit number (e.g. 42760054418). Given the codes and the names of the 50 Ankara branches of XYZ Bank in file "Branches.txt", and 200 account numbers and the money in each account in file "Accounts.txt", separate the accounts opened in Etlik Branch and those opened in Cubuk Branch, into two arrays, and merge the accounts opened in Etlik and Cubuk branches, in ascending order of the money in those accounts, and display the result.

8) Given a 50 x 5 matrix, consisting of 5 quiz grades of 50 students, find and display the fourth quiz grade of the student who got 25 from the third quiz, if any.

9) Find the position of the last occurrence of a double number in an ascendingly sorted one-dimensional array, using linear search algorithm, recursively.

10) Find the greatest common divisor of two positive integers. The greatest common divisor of two positive integers M and N, GCD (M, N), is the largest positive integer that divides both M and N. Thus, for instance, GCD (100, 5) == 5, GCD (100, 33) = 1, GCD (102, 30) = 6.

This can be found by using the division algorithm, as follows:

$$102 = 30 * 3 + 12$$

$$30 = 12 * 2 + 6$$

$$12 = 6 * 2 + 0$$

Notice that:

$$\begin{aligned} \text{GCD}(102, 30) &= \text{GCD}(30, 12) \\ &= \text{GCD}(12, 6) \\ &= 6 \end{aligned}$$

In each case, the remainder is used in the next step. The process terminates when a remainder of 0 is obtained. Write a modular C code for this purpose.

11) Write recursive functions that display each of the following outputs when they are called with the string on the first line.

a) THE EXAM  
HE EXAM  
E EXAM  
EXAM  
EXAM  
XAM  
AM  
M

b) THE EXAM IS VERY EASY  
HE EXAM IS VERY EAS  
E EXAM IS VERY EA  
EXAM IS VERY E  
EXAM IS VERY  
XAM IS VERY  
AM IS VER  
M IS VE  
IS V  
IS  
S

12) Define a date structure consisting of day, month, year. Define another structure consisting of the name, surname, and birthdate of a person. Define a function that decides whether a certain date is less than another date. Write a program that reads the information of your friends from a text file **friends.txt**, and creates two binary files **younger.bin** and **older.bin** containing your friends whose are older and younger than you, respectively, and displays on the screen the name and surname of your friends who have born in the same date as you, if any. Define a function for displaying the content of a binary file containing persons so that you can check your work.

13) Write a program that read the binary file **others.bin** containing a list of integers, and writes the prime numbers into another binary file **primes.bin**, making use of a recursive **is\_prime** function. Use the function **disp\_bin** to display the content of **primes.bin** so that you can check your work.

14) Define a student structure consisting of id, name, surname, and cgpa. Write a program that reads the information about some students from a text file whose name is given as input, validates the file name, and copies them into a binary file, with the same name but with the extension **.bin**. The program then displays a menu with the options:

- 1) Go to student X from top
- 2) Move X students ahead
- 3) Go X students back from bottom
- 4) Exit

and performs the chosen operation, by taking the value of X as a positive integer, and displaying the information of the student reached after the operation.