

**Programming and Data Structures Laboratory | 2021-22 Autumn semester, Section 20**  
**Assignment 7 | February 15, 2022**

**Submission instructions**

\* Submit one .c file for each part of the assignment. Each of your .c files should be named as:

**<your roll number>\_A<assignment number>\_<part number>.c**

For instance, if your roll number is 21CS10023, and if you are presently doing Assignment 7 which has 3 parts, then you should submit 3 separate .c files named as:

21CS10023\_A7\_1.c    21CS10023\_A7\_2.c    21CS10023\_A7\_3.c

\* Submissions must be through the course Moodle, before the end of Lab session (11:55 AM). Late submissions will be penalized / not accepted.

---

**1. [15 marks]** Take two strings s1 and s2 as inputs from the user. Assume that s1 and s2 are sequences of English words. Combine s1 and s2 into a string s of words as follows: Insert words alternately from s1 and s2 in s, starting with s1. If all the words of one of the strings s1 or s2 have already been inserted into s, copy the rest of the other string into s. Finally print s on the screen.

**Example interaction with user:**

Enter s1: It is raining outside

Enter s2: Ram has gone to market

It Ram is has raining gone outside to market

**2. [15 marks]** Write a C program that stores the following 10 strings using a 2-dimensional character array: "messi", "ronaldo", "beckham", "romario", "klose", "ronaldinho", "kaka", "maradona", "pele", and "zidane". Each row of the 2-d array should store one string.

Then the program should ask the user for inputting one character. The program should search for the input character in all strings, and print out that string which contains the input character most number of times. For instance, if the input character is 's', the program should print out "messi"; if the input is 'k', the program should print out "kaka". In case there are multiple strings which contain the input character the maximum number of times (i.e., in case of a tie), then all those strings should be printed out, one in each line.

**3. [20 marks]** Write a program that helps a user to maintain a dynamic set of integers. The program should start with an empty integer array of size 100, which will be used to store the set. The user should be given options to (i) insert numbers from this set, (ii) delete numbers from this set, (iii) display the set in set notation (i.e., within curly braces, and elements separated by comma), and (iv) exit from the program. When the user wants to insert a number, the program should check whether the number already exists in the set (every element of a set should be distinct). If yes, the program should inform the user; otherwise the element should be inserted into the array. Again, when the user wants to delete

a number, the program should check whether the number exists in the set. If not, the user should be informed; otherwise the said number should be deleted from the set.

There should be separate functions for inserting a number into the set, deleting a number from the set, and for displaying the set.

Throughout the operations, the array should be maintained sorted in ascending order, and binary searching should be used to check for the presence of numbers. It will be helpful if you write a search function (using binary search) to detect if a certain element is contained in the set; this function can be used both within the insert and delete functions.

An example interaction between the user (U) and the program (P) is as follows:

P: Press 1 to insert numbers, 2 to delete numbers, 3 to display the set, and 0 to exit.

U: 1

P: Enter number to insert:

U: 43

P: 43 inserted [Note: 43 should be placed in the first element of the array]

P: Press 1 to insert numbers, 2 to delete numbers, 3 to display the set, and 0 to exit.

U: 1

P: Enter number to insert:

U: 21

P: 21 inserted [Note: 21 should be placed in the first element of the array, and 43 should be shifted to the second position.]

P: Press 1 to insert numbers, 2 to delete numbers, 3 to display the set, and 0 to exit.

U: 1

P: Enter number to insert:

U: 73

P: 73 inserted [Note: 73 should be placed in the third element of the array]

P: Press 1 to insert numbers, 2 to delete numbers, 3 to display the set, and 0 to exit.

U: 1

P: Enter number to insert:

U: 43

P: Number already exists

P: Press 1 to insert numbers, 2 to delete numbers, 3 to display the set, and 0 to exit.

U: 3

P: {21, 43, 73}

P: Press 1 to insert numbers, 2 to delete numbers, 3 to display the set, and 0 to exit.

U: 2

P: Enter number to delete

U: 43

P: 43 deleted [Note: 73 should be brought to the second element of the array]

P: Press 1 to insert numbers, 2 to delete numbers, 3 to display the set, and 0 to exit.

U: 2

P: Enter number to delete

U: 50

P: This number does not exist in the set

P: Press 1 to insert numbers, 2 to delete numbers, 3 to display the set, and 0 to exit.

U: 0