CSE108
LW 09

- **Using mobile phones, flash disks, internet and any other record or communication media is strictly forbidden during lab sessions. Throughout a lab session, all such media must be kept turned off and in a closed environment. Violation of this rule is punished with a grade 0, -100 or worse. Before doing anything else, make sure that your computer is not attached any such media.**
- **Make sure that you have deleted all of your work PARMANENTLY before leaving the first sessions.**

In the first 2 parts of this lab work you will write a program to determine people living in a given location using the arrays described below. You will initialize these arrays in main().

names: A string array including the names and surnames of people we interest.
locations: A string array including cities people in “names” array lives.
names_locations: An integer array that corresponds people in “names” and cities in “locations”. Each entry in this array is an integer parallel with “names” such that nth entry in this file indicates the index of the location of the nth entry of the array “names”. For example, in the following case, Ali and Mehmet live in Istanbul, Melike lives in Ankara, Ahmet and Selen lives in Izmir.

names
Ali Veli Ahmet Selvi Melike Selen Mehmet Yigit Selen Karayel

names_locations
0 2 1 0 2

Locations
Istanbul Ankara Izmir

PART-I (1 PT)

Write the following **recursive** function to print the contents of the above-defined string arrays:

- **print_strings**: takes a string array and prints its content.

PART-II (3 PT)

Write a function which takes the arrays including the entries of the above defined text files and a location (as a string) and returns people living in that location in a string array as an output argument. Test your implementation.

PART-III (3 PT)

Implement the following **recursive** functions. You are not allowed to change the function prototype, use global or static variables.

- **int find_first_occurrence(const char str[], char ch)**: **(1Pt)** returns the index of the first occurrence of the character in the string if exists, and returns -1 otherwise. e.g., find_first_occurance(“d**e**neme”, '**e**') should return 1, find_first_occurance(“deneme”, 'a') should return -1.

- **int find_last_occurance(const char str[], char ch)**: **(2Pt)** returns the index of the last occurrence of the character in the string if exists, and returns -1 otherwise. e.g., find_last_occurance(“denem**e**”, '**e**') should return 5,

PART-IV (1 PT)

Implement the following recurrence relation using a recursive function:

$$\boldsymbol{f(x)} = \begin{cases} f(x, y-1) + f(x-1, y) + x + y, & if\ x > 0\ and\ y > 0 \\ 0, & otherwise \end{cases}$$

PART-BONUS (2 PT)

In this part, you will generate an array of pointers which shows the starting adress of the second words in a string array. E.g., if you use the names array as the string array, n-th row of the array of pointer you generated should show the surname of the n-th entry. Implement the following functions:

- **get_ptr_arr**: a function to generate the pointer array
- **print_ptr_arr**: a function to print the content of the pointer array.
- **modify_ptr_arr**: a function to change the content of a given index of a pointer array

In test:

- print the names array
- generate a pointer array from names
- print the pointer array
- change the 2nd index of the pointer array
- print the names array
- print the pointer array (observe that change affects both array)