Guided Exercise 03

Linked lists & Memory Management
Open Book; Open Notes; Time Given =120 minutes

“By proceeding I certify that I have neither received nor given unpermitted aid on this examination and that I have reported all such incidents observed by me in which unpermitted aid is given.”

Zip the assignment folder and rename it to ’<rollnumer>_GE4.zip’. Upload your zip file in the corresponding LMS submission tab. Example, ’26100181_GE4.zip’

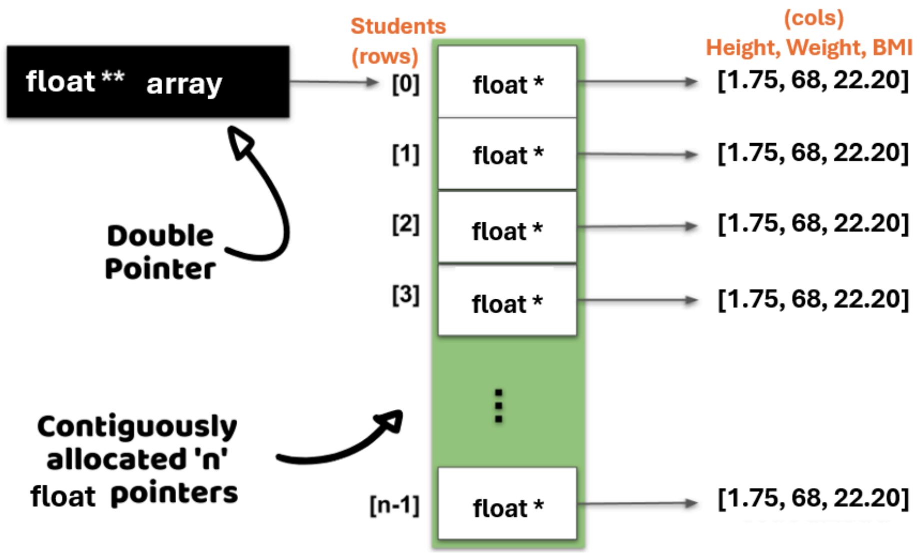
Task 1: 2D Dynamic Arrays (marks = 20).

In this lab, you will learn essential skills in memory management, including how to dynamically allocate and deallocate memory for a 2D dynamic array and, accessing and manipulating its elements.

Your task is to implement the following functions:

- **float** allocateArray(int students) {**

This function dynamically allocates a 2D array using malloc(). Each row represents a student, and the three columns store the student's height, weight, and BMI. The array will have a fixed column size of 3. The function returns a double pointer to the allocated memory. The diagram below shows the structure:



- **void deallocateArray(float** arr, int students)**

This function deallocates a dynamically allocated 2D array. It takes a double pointer to the array (arr) and the number of students (rows) as parameters.
Note: free() can only deallocate one dynamically allocated block of memory at a time. If you have a dynamically allocated 2D array, you need to free() each row or dynamically allocated block individually before freeing the array of pointers itself. Each malloc or calloc call needs a corresponding free() call to release the memory.

- **void calculateBMI(float** arr, int students)**

This function calculates the BMI for each student and stores it in the third column of the 2D array. The array *arr* has *students* rows, with each row holding a student's height, weight, and BMI. The function loops through each student, retrieves their height and weight, calculates BMI using the formula: $BMI = weight/ height^2$, and stores the result in the third column.

- **int countStudentsInBMIRange(float** arr, int students, float low, float high)**

This function counts how many students have a BMI within a specified range (low to high). It loops through the BMI values stored in the third column of the 2D array. For each student, if their BMI is within the given range, the count is incremented. The function returns the total count.

Task 2: Recursion (marks = 30)

- **Max of Array**

Given an array of integers, write a recursive function that returns the maximum value in that array.

- **Greatest Common Divisor**

Given two integers x and y, the following recursive definition determines the greatest common divisor of x and y, written gcd(x,y):

$$\gcd(x,y) = \begin{cases} x & \text{if } y = 0 \\ \gcd(y, x\%y) & \text{if } y \neq 0 \end{cases}$$

Write a recursive function, gcd, that takes as parameters two integers and returns the greatest common divisor of the numbers.

- **Palindrome**

A palindrome is a string that reads the same both forward and backward. For example, the string "madam" is a palindrome. Write a recursive function to check whether a string is a palindrome. Your function should return true if the string is a palindrome and false otherwise.