

PDS Lab, Section - 17, Date: 15th Feb 2021

Assignment - 8 [Multi-Dimensional Arrays, Dynamic Memory Allocation (DMA), Sorting & Searching]

Instructions

1. Create a directory named as <rollno>_A8, where <rollno> is your roll number.
2. Give the name of the program as <p>.c where <p> implies the problems number, like 1.c 2.c 3.c etc. Store all the program under this Assignment in the directory <rollno>_A8
3. Zip the entire directory <rollno>_A8.
4. You should upload your zipped file <rollno>_A8.zip to the Moodle course web page latest by 5:30 PM (without penalty). The cutoff time will be till 6:00 PM with a penalty of 25% on your secured marks (i.e., if you secured 80 marks, after penalty you will get 60 marks). Beyond 6:00 PM, the moodle system will not allow you to submit, as a result you will get zero.

-
- 1) Write a C program using dynamic memory allocation to perform the following:
 - a) Enter an integer n and allocate memory for n number(s) of int variable(s).
 - b) Enter n elements and print them.
 - c) Enter new size, and reallocate memory accordingly. If the new size is greater than n, then additional numbers (newsize - oldsize) are entered by user after reallocation.
 - d) Print all the elements after reallocation.**[15 Marks]**

Example:

Input : Enter the size of list : 5

Enter the integers into list: 1 2 3 4 5

The numbers in the list are: 1 2 3 4 5

Case1: Enter the new size of the list: 8

Enter 3 numbers: 6 7 8

Output : The numbers in the list are: 1 2 3 4 5 6 7 8

Case2: Enter the new size of the list: 3

Output : The numbers in the list are: 1 2 3

- 2) Write a C program to dynamically allocate a 2D Matrix for which the user inputs the dimensions of the matrix.
- a) Write a function to enter the elements of a matrix and return the matrix to main function.
 - b) Write a function to print the elements of a matrix
 - c) Write a function to sort the elements of a matrix in ascending order and return the sorted matrix to main ().
 - d) Print the elements of a sorted matrix.
 - e) Write a function to search the given key (through keyboard) row-wise using binary search and return the key position in the matrix using row-col indices.

Example :

[25 Marks]

Dimensions of matrix :

Rows = 3

Columns = 3

Enter the elements of a matrix

4 6 2

8 1 3

5 9 6

Matrix after arranging rows in ascending order

2 4 6

1 3 8

5 6 9

Search the key value in sorted matrix:

Key = 12








Key is not present in the matrix

Key = 6

Key is present in the matrix

Position of the key = (3,2)

- 3) Define a structure called House which has an integer variable color that is either 0 or 1 or -1, and another integer value that is the age of the house (how old the house is). Also there is a geographical map that is divided into 2d matrix grids of size N X N, where each grid either has a house in it or not . As shown in the figure below **[30 Marks]**

If the value of color is 1, it is an orange house, if it is 0 it is a blue house and if the value of color is -1, there is no house at that position in the grid.

- Take N as an input from the user, and create a 2d Matrix of size N*N of the data type House using DYNAMIC MEMORY ALLOCATION and take the input from the user, if the user enters the value of color to be -1 , no need to take the Age of that house as input.
- Write a function find_correct_house(), which takes in the 2d matrix as argument and row size N, and counts the number of BLUE houses such that they are surrounded by at least 2 ORANGE Houses and return that number

- 4) Write a C program using pointers and dynamic memory allocation to perform the following:
- (a) Enter the list of words through keyboard, where the size of the list N needs to be specified by the user. Allocate the memory for each word as per its size.
 - (b) Once the list is available, sort the list.
 - (c) Split the sorted list into 2-D array based on the 1st character of the word.

Example:

[30 Marks]

Size of the list = 15

List of words = arabic, apple, xerox, puppy, box, baby, xxx, cat, cot, sun, cup, doctor, elephant, danger, public

List of sorted words = apple, arabic, baby, box, cat, cot, cup, danger, doctor, elephant, public, puppy, sun, xerox, xxx

List of word in 2-D array =

apple, arabic

baby, box

cat, cot, cup

danger, doctor

elephant

public, puppy

sun

xerox, xxx