CS19001: Programming and Data Structures Laboratory

02-April-2025

SD DRC SB

Department of Computer Science & Engineering,
Indian Institute of Technology Kharagpur,
Paschim Medinipur, West Bengal, India – 721302.

Programming Assignments

Complete and submit during lab

Assignment 1: *Sparse vector operations*

Objective:

Implement a sparse vector data type in C that supports input, output, and addition operations.

Background:

A sparse vector is a vector in which most elements are zero. Instead of storing all elements, we only store the nonzero values and their corresponding indices.

For example, the vector:

V=[0,0,5,0,0,8,0,0,3]

Can be stored as:

(index, value)={(2,5),(5,8),(8,3)}(index, value)={(2,5),(5,8),(8,3)}

Assignment 1: *Sparse vector operations*

Task Requirements:

You must implement the following:

Structure Definition (SparseVector):

Use a struct to store nonzero elements as (index, value) pairs. Use dynamic memory allocation to store elements efficiently.

Functions to Implement:

`SparseVector* read_vector(int size)`: Reads a sparse vector from user input.

`void print_vector(SparseVector* vec)`: Prints the sparse vector.

`SparseVector* add_vectors(SparseVector* v1, SparseVector* v2)`: Adds two sparse vectors.

Main Function: Read two sparse vectors from input. Print both vectors. Compute and print their sum.

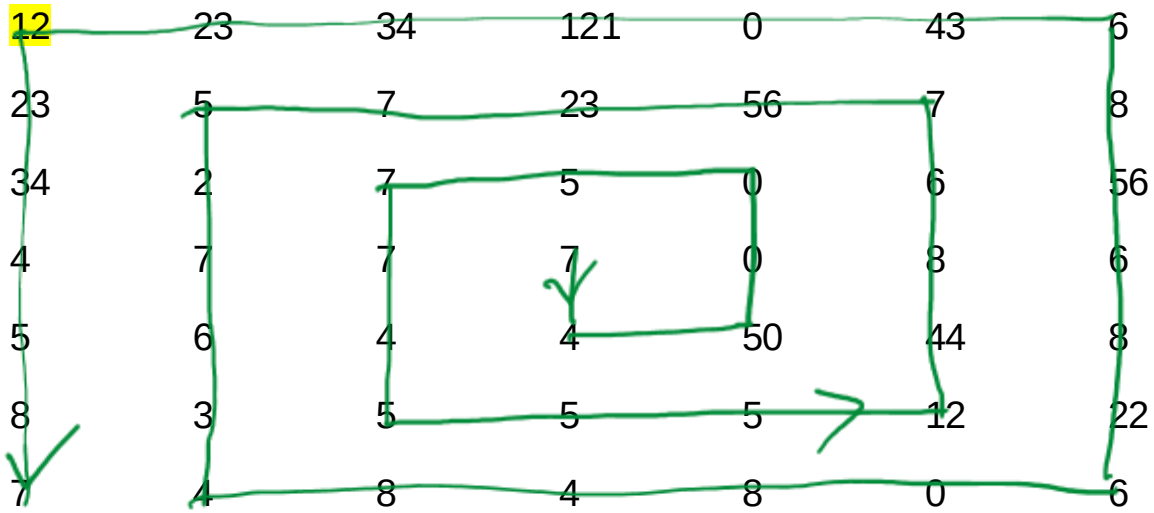
Print all input and output in both 2-d format as well as output format.

Assignment 2: *2-d Traversal*

Your program takes as input a 2d array of dimension (2n+1)*(2n+1)

The input is taken row wise, i.e. user inputs all elements in one row (separated by space) and then presses enter for entering next row.

Your program makes the following anti-clockwise traversal from center and prints the elements in that order.



O/P: 7, 4, 50, 0, 0, 5, 7, 7, 4 ……

Assignment 3: 2-d routing

Your program takes as input a 2d array of dimension (2n+1)*(2n+1)

The input is taken row wise, i.e. user inputs all elements in one row with space and then presses enter for entering next row. Each cell contains a 1 with a few cells containing a 0 (MINES)

						END
				0		↑
						↑
				0	0	↑
0	0	0	0			↑
			→	→	→	↑
START	→	→	↑	0		

Your program prints ALL paths from START to END (recursion?) in the input 2-d square . MINES are to be avoided. The example path given will be printed as :

R, R, UP,R, R, R, UP, UP, UP, UP, UP.

Thank You!