1.Write a C program to implement a Directed/Undirected Graph using adjacency matrix and show all the following operations in switch case, i) Create a graph ii) Display graph

Code:

```c
#include <stdio.h>
#include <malloc.h>

void main() {
    int n;

    printf("Enter the number of nodes: ");
    scanf("%d", &n);

    int arr[n][n];

    for(int i = 0; i < n; i++) {
        for(int j = 0; j < n; j++) {
            printf("%d x %d: ", i+1, j+1);
            scanf("%d", &arr[i][j]);
        }
    }

    printf("\nAdjacency Matrix Representation: \n");
    for(int i = 0; i<n; i++){
        for(int j = 0; j<n; j++){
            printf("%d ", arr[i][j]);
        }
        printf("\n");
    }
    printf("\nAdjacency List Representation: \n");
    for(int i = 0; i < n; i++)
    {
        printf("%d ", i+1);
        int flag = 0;
        for(int j = 0; j < n; j++) {
            if (arr[i][j] != 0) {
                printf("-> %d", j+1);
                flag = 1;
            }
        }
        if (flag == 0) {
            printf("-> X");
        }
        printf("\n");
    }
}
```

Output:

```
PS C:\Users\Ajay kumar\Desktop\SEIT-B\DSA\Lab\4 - Graph> cd
"c:\Users\Ajay kumar\Desktop\SEIT-B\DSA\Lab\4 - Graph\" ; if ($?) {
gcc main.c -o main } ; if ($?) { .\main }

Enter the number of nodes: 4
1 x 1: 1
1 x 2: 0
1 x 3: 0
1 x 4: 1
2 x 1: 1
2 x 2: 1
2 x 3: 0
2 x 4: 0
3 x 1: 1
3 x 2: 0
3 x 3: 1
3 x 4: 0
4 x 1: 1
4 x 2: 1
4 x 3: 1
4 x 4: 1

Adjacency Matrix Representation:
1 0 0 1
1 1 0 0
1 0 1 0
1 1 1 1

Adjacency List Representation:
1 -> 1-> 4
2 -> 1-> 2
3 -> 1-> 3
4 -> 1-> 2-> 3-> 4
PS C:\Users\Ajay kumar\Desktop\SEIT-B\DSA\Lab\4 - Graph>
```

A. Explain Directed and Weighted Graph

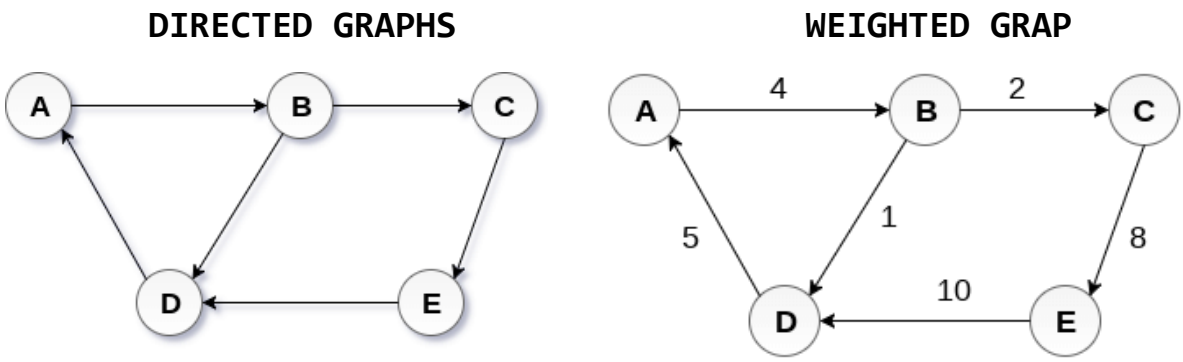
Answer:

Directed Graph (Digraph):

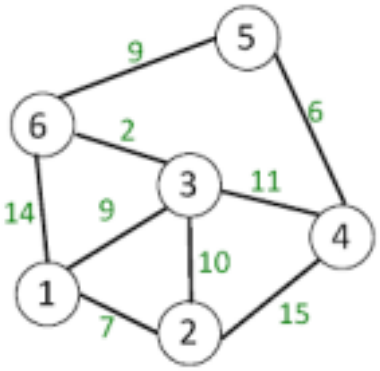
- A directed graph, also known as a digraph, is a type of graph in which the edges have a direction associated with them. This means that the relationship between nodes (vertices) is one-way, indicating that you can only traverse from one node to another in a specific direction.
- In a directed graph, each edge has an associated source vertex (starting node) and a target vertex (ending node). The edge goes from the source to the target.

Weighted Graph:

- A weighted graph is a type of graph in which each edge has an associated numerical value called a weight or cost. These weights represent some quantity such as distance, cost, time, or any other relevant measure associated with the connection between nodes.
- Weighted graphs are useful for modeling scenarios where not all connections between nodes are equal, and you want to take into account the relative importance or cost of each edge.
- Weighted graphs are commonly used in applications like route planning (finding the shortest or cheapest path between two nodes), network flow optimization, and spanning tree algorithms (finding minimum spanning trees).



B.Show the adjacency matrix and adjacency list representation for the graph given below.



Answer:

Adjacency Matrix Representation:

	1	2	3	4	5	6
1	0	7	9	0	0	14
2	7	0	10	15	0	0
3	9	10	0	11	0	2
4	0	15	11	0	6	0
5	0	0	0	6	0	9
6	14	0	2	0	9	0

Adjacency List Representation: