**9.a Depth First Search:**

#include <stdio.h>

#define MAX 5

void depth\_first\_search(int adj[][MAX], int visited[], int start, int n)

{

int stack[MAX];

int top = -1,i;

printf("%c ", start + 65);

visited[start] = 1;

stack[++top] = start;

while (top !=-1)//Use Stack

{

start = stack[top];

for (i = 0; i < n; i++)

{

if (adj[start][i] && visited[i] == 0)

{

stack[++top] = i;

printf("%c ", i + 65);

visited[i] = 1;

break;

}

}

if (i == n)

top--;

}

}

int main()

{

int adj[MAX][MAX];

int visited[MAX] = {0}, i , j, n;

printf("How many elements \n");

scanf("%d",&n);

printf("\n Enter adjecency matrix: \n");

for(i=0;i<n;i++)

for(j=0;j<n;j++)

scanf("%d",&adj[i][j]);

printf("DFS Traversal: \n");

depth\_first\_search(adj, visited, 0,n);

printf("\n");

return 0;

}

How many elements

5

Enter adjecency matrix:

0 1 0 1 0

1 0 1 1 0

0 1 0 0 1

1 1 0 0 1

0 0 1 1 0

DFS Traversal:

A B C E D

**9.b Breadth First Search**

**Program:**

#include <stdio.h>

#define Max 5

void bfs(int adj[][Max],int visited[],int start, int n)

{

int queue[Max],rear=-1,front=-1,i;

queue[++rear]=start;

visited[start]=1;

while(rear !=front)

{

start=queue[++front];

printf("%c \t",start + 65);//print traversed element

for(i=0;i<n;i++)

{

if(adj[start][i]==1 && visited[i]==0)

{

queue[++rear]=i;

visited[i]=1;

}

}

}

}

int main ()

{

int visited[Max]={0};

int adj[Max][Max],i,j,n;

printf("How many elements");

scanf("%d",&n);

printf("\n Enter adjecency matrix: \n");

for(i=0;i<n;i++)

for(j=0;j<n;j++)

scanf("%d",&adj[i][j]);

printf("BFS Traversal: \n");

bfs(adj,visited,0,n);

return 0;

}

**Output:**

How many elements: 5

Enter adjecency matrix:

0 1 0 1 0

1 0 1 1 0

0 1 0 0 1

1 1 0 0 1

0 0 1 1 0

BFS Traversal:

A B D C E