**Implementation of BSF**

#include <stdio.h>

#include <stdlib.h>

#define MAX 10

int n;

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

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

queue[++rear] = start;

visited[start] = 1;

printf("BFS Traversal : ");

while(rear != front) {

start = queue[++front];

if(start == 4) {

printf("5\t");

}

else {

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

}

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;

printf("Enter number of vertices : ");

scanf("%d", &n);

printf("\nEnter the adjacency matrix : \n");

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

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

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

}

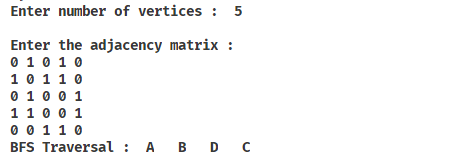
}

bfs(adj,visited,0);

return 0;

}

// output



**Implementation of DFS**

#include <stdio.h>

#include <stdlib.h>

int n;

void dfs(int adj[][n],int visited[],int start) {

int stack[n];

int top = -1, i;

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

visited[start] = 1;

stack[++top] = start;

while(top != -1) {

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[10][10];

int visited[20] = {0}, i, j;

printf("Enter number of vertices : ");

scanf("%d", &n);

printf("\nEnter the adjacency matrix: \n");

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

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

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

}

}

printf("DFS Traversal : ");

dfs(adj,visited,0);

return 0;

}

// output

