NAME: TANISHA C A
REGISTER NO.:230701390

Ex-11: Implementation of BFS and DFS

```
#include<stdio.h>
#include<stdlib.h> #define
size 7 int s[size]; int
top=-1; int pop(); void
push(int); int
queue[size]; int front = -
1, rear = -1; void dfs();
void bfs();

int isEmpty() { return front == -1 && rear == -1; }

int isFull() { return rear == size - 1; }
 void enqueue(int val) {
if (!isFull()) {         if
(isEmpty()) {
front = rear = 0;
        } else {
            rear = (rear + 1) % size;
        }
        queue[rear] = val;
    } else {
        printf("\nQUEUE IS FULL!\n");
    }
}  int dequeue() {      if
(!isEmpty()) {          int val
= queue[front];          if
(front == rear) {
front = rear = -1;
        } else {
            front = (front + 1) % size;
        }
return val;     }
else {
        printf("\nQUEUE IS EMPTY!\n");
return -1;
    }
}  void
dfs(){
int
g[size][size]={{0,1,1,0,0,0,0},{0,0,0,0,0,0,0},{0,0,0,1,0,1,0},{1,1,0,0
,0,0,1},{0,1,0,0,0,0,0},{0,0,0,0,0,0,1},{0,0,0,0,1,0,0}};
int visited[size]={0};     int j,i=0;

   printf("DFS :  ");
while(i>-1 && i<size)
    {
        if(visited[i]!=1)
{         printf("%d-
>",i);
visited[i]=1;         }
        for(i,j=0;j<size;j++)
        {
            if(g[i][j]==1 && visited[j]!=1){
push(j);
            }
}
i=pop();
    }

}  void bfs(){     int
g[size][size] = {         {0,
1, 1, 0, 0, 0, 0},
        {0, 0, 0, 0, 0, 0, 0},
        {0, 0, 0, 1, 0, 1, 0},
        {1, 1, 0, 0, 0, 0, 1},
        {0, 1, 0, 0, 0, 0, 0},
        {0, 0, 0, 0, 0, 0, 1},
        {0, 0, 0, 0, 1, 0, 0}
    };
    int visited[size]={0};
     int i = 0;
printf("BFS : ");
visited[i] = 1;
printf("%d->", i);
     enqueue(i);
while (!isEmpty()) {
int i = dequeue();
         for (int j = 0; j < size; j++) {
if (g[i][j] && !visited[j]) {
visited[j] = 1;
printf("%d->", j);
enqueue(j);
            }
        }
    }

}

void push(int data)
{         top=top+1;
s[top]=data; }  int
pop() {         int
temp;
temp=s[top];
top=top-1;
return temp;

}  int main() {
int ch,ans=1;
do{
    printf("enter your choice \n1.DFS\n2.BFS\n");
scanf("%d",&ch);     switch(ch)     {
case 1:         dfs();         break;
case 2:         bfs();         break;     }
    printf("\nWant to continue ?\n1.yes \n0.no\n");
scanf("%d",&ans);
    }
while(ans==1);
}
```