# LAB 1

AIM:-To find the dfs and bfs by using adjacent matrix
Code:-

```
#include<stdio.h>

int q[20],top=-1,front=-1,rear=-1,a[20][20],vis[20],stack[20];
int delete();
void add(int item);
void bfs(int s,int n);
void dfs(int s,int n);
void push(int item);
int pop();

void main()
{
int n,i,s,ch,j;
char c,dummy;
printf("ENTER THE NUMBER VERTICES ");
scanf("%d",&n);
for(i=1;i<=n;i++)
{
for(j=1;j<=n;j++)
{
printf("ENTER 1 IF %d HAS A NODE WITH %d ELSE 0 ",i,j);
scanf("%d",&a[i][j]);
}
}
printf("THE ADJACENCY MATRIX IS\n");
for(i=1;i<=n;i++)
{
for(j=1;j<=n;j++)
{
printf(" %d",a[i][j]);
}
printf("\n");
}

do
{
for(i=1;i<=n;i++)
vis[i]=0;
printf("\nMENU");
```

```
printf("\n1.B.F.S");
printf("\n2.D.F.S");
printf("\nENTER YOUR CHOICE");
scanf("%d",&ch);
printf("ENTER THE SOURCE VERTEX :");
scanf("%d",&s);

switch(ch)
{
case 1:bfs(s,n);
break;
case 2:
dfs(s,n);
break;
}
printf("DO U WANT TO CONTINUE(Y/N) ? ");
scanf("%c",&dummy);
scanf("%c",&c);
}while((c=='y')||(c=='Y'));
}

void bfs(int s,int n)
{
int p,i;
add(s);
vis[s]=1;
p=delete();
if(p!=0)
printf(" %d",p);
while(p!=0)
{
for(i=1;i<=n;i++)
if((a[p][i]!=0)&&(vis[i]==0))
{
add(i);
vis[i]=1;
}
p=delete();
if(p!=0)
printf(" %d ",p);
}
for(i=1;i<=n;i++)
if(vis[i]==0)
bfs(i,n);
}
```

```
void add(int item)
{
if(rear==19)
printf("QUEUE FULL");
else
{
if(rear==-1)
{
q[++rear]=item;
front++;
}
else
q[++rear]=item;
}
}
int delete()
{
int k;
if((front>rear)||(front==-1))
return(0);
else
{
k=q[front++];
return(k);
}
}

void dfs(int s,int n)
{
int i,k;
push(s);
vis[s]=1;
k=pop();
if(k!=0)
printf(" %d ",k);
while(k!=0)
{
for(i=1;i<=n;i++)
if((a[k][i]!=0)&&(vis[i]==0))
{
push(i);
vis[i]=1;
}
```

```
k=pop();
if(k!=0)
printf(" %d ",k);
}
for(i=1;i<=n;i++)
if(vis[i]==0)
dfs(i,n);
}
void push(int item)
{
if(top==19)
printf("Stack overflow ");
else
stack[++top]=item;
}
int pop()
{
int k;
if(top==-1)
return(0);
else
{
k=stack[top--];
return(k);
}
}
```

Out put:-

"C:\Users\Admin\Desktop\1BM21CS059\2ND ADA LAB\LAB2\bin\Debug\LAB2.exe"

```
ENTER THE NUMBER VERTICES 4
ENTER 1 IF 1 HAS A NODE WITH 1 ELSE 0 0
ENTER 1 IF 1 HAS A NODE WITH 2 ELSE 0 1
ENTER 1 IF 1 HAS A NODE WITH 3 ELSE 0 1
ENTER 1 IF 1 HAS A NODE WITH 4 ELSE 0 1
ENTER 1 IF 2 HAS A NODE WITH 1 ELSE 0 0
ENTER 1 IF 2 HAS A NODE WITH 2 ELSE 0 0
ENTER 1 IF 2 HAS A NODE WITH 3 ELSE 0 0
ENTER 1 IF 2 HAS A NODE WITH 4 ELSE 0 1
ENTER 1 IF 3 HAS A NODE WITH 1 ELSE 0 0
ENTER 1 IF 3 HAS A NODE WITH 2 ELSE 0 0
ENTER 1 IF 3 HAS A NODE WITH 3 ELSE 0 0
ENTER 1 IF 3 HAS A NODE WITH 4 ELSE 0 0
ENTER 1 IF 4 HAS A NODE WITH 1 ELSE 0 0
ENTER 1 IF 4 HAS A NODE WITH 2 ELSE 0 0
ENTER 1 IF 4 HAS A NODE WITH 3 ELSE 0 1
ENTER 1 IF 4 HAS A NODE WITH 4 ELSE 0 0
THE ADJACENCY MATRIX IS
 0 1 1 1
 0 0 0 1
 0 0 0 0
 0 0 1 0

MENU
1.B.F.S
2.D.F.S
ENTER YOUR CHOICE1
ENTER THE SOURCE VERTEX :1
 1 2  3  4 DO U WANT TO CONTINUE(Y/N) ? y

MENU
1.B.F.S
2.D.F.S
ENTER YOUR CHOICE2
ENTER THE SOURCE VERTEX :1
 1  4  3  2 DO U WANT TO CONTINUE(Y/N) ?
```

Notes:-

LAB-2 15th June.

Aim:- To find out the DFS and BFS by using Adjacent matrix.

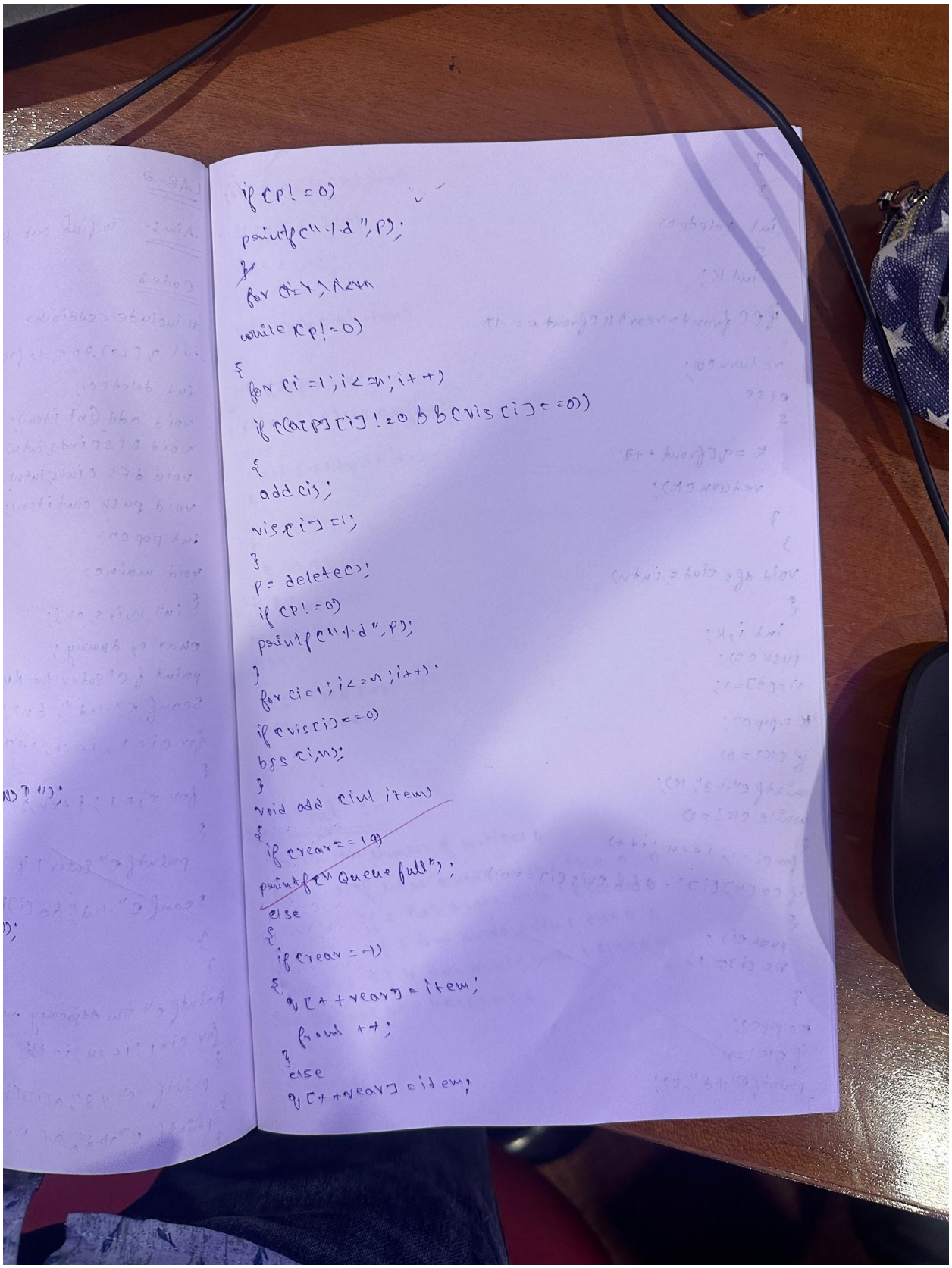
Code:-

```
#include <stdio.h>
int q[20],top=-1,front=-1,rear=-1,a[20][20],stack[20];
int delete();
void add(int item);
void bfs(int s,int n);
void dfs(int s,int n);
void push(int item);
int pop();
void main()
{ int n,i,s,ch,j;
char c,dummy;
printf("Enter the Number vertices");
scanf("%d",&n);
for(i=1;i<=n;i++)
{
 for(j=1;j<=n;j++)
{
 printf("Enter 1 if %d has a Node with %d else 0",i,j);
scanf("%d",&a[i][j]);
}
}
printf("The Adjacency matrix is\n");
for(i=1;i<=n;i++)
{
 printf("%d",a[i][j]);
}printf("\n");
}
```

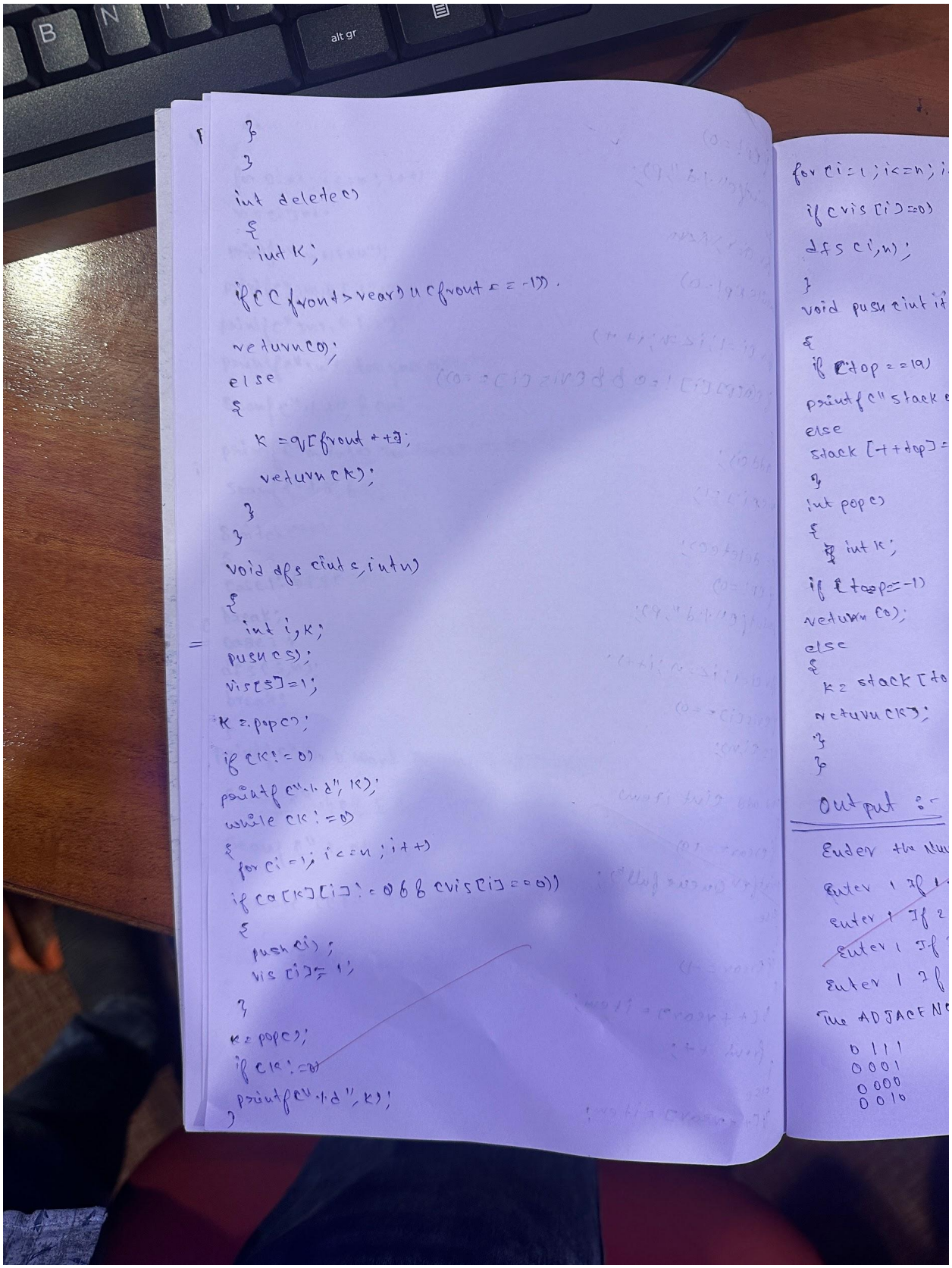
```
do
{
for (i=1; i<=n; i++)
vis[i]=0;
printf("\n1.BFS");
printf("\n2.DFS");
printf("\n Enter your choice");
scanf("%d", &ch);
printf("\n Enter the source vertex:");
scanf("%d", &s);
switch(ch)
{
case 1: bfs(s,n);
break;
case 2:
dfs(s,n);
break;
}
printf("\n Do u want to continue (Y/N)? ");
scanf("%d", &dummy);
scanf("%c", &c);
} while((c=='y')||(c=='Y'));

void bfs(int s,int n)
{
int p,i;
add(s);
vis[s]=1;
p=delete();
```

```
if(p!=0)
printf("%d", p);
for(i=1;i<=n
while(p!=0)
{
for(i=1;i<=n;i
if(a[p][i]!=0
{
add(i);
vis[i]=1;
}
p=delete();
if(p!=0)
printf("%d", p
}
for(i=1;i<=n;i
if(vis[i]==0)
bfs(i,n);
}
void add(int it
{
if(rear==(q
printf("Queue f
else
{
if(rear==-1)
{
q[++rear]=
front++;
}
else
q[++rear]=
```



```
if(p!=0)
printf("%d ",p);
}
for(i=1;i<n
while(p!=0)
{
for(i=1;i<=n;i++)
if((a[p][i]!=0 && (vis[i]==0))
{
add(i);
vis[i]=1;
}
p=delete();
if(p!=0)
printf("%d",p);
}
for(i=1;i<=n;i++)
if(vis[i]==0)
bfs(i,n);
}
void add(int item)
{
if(rear==19)
printf("Queue full");
else
{
if(rear==-1)
{
q[++rear]=item;
front++;
}
else
q[++rear]=item;
```



```
}
}
int delete()
{
 int k;
if((front>rear)||(front==-1))
return(0);
else
{
  k=q[front++];
  return(k);
 }
}
void dfs(int s,int n)
{
 int i,k;
push(s);
vis[s]=1;
k=pop();
if(k!=0)
printf("%d ",k);
while(k!=0)
{
for(i=1;i<=n;i++)
if((a[k][i]!=0)&&(vis[i]==0))
{
 push(i);
 vis[i]=1;
}
k=pop();
if(k!=0)
printf("%d ",k);
}
```

```
for(i=1;i<=n;i
if(vis[i]==0)
dfs(i,n);
}
void push(int i
{
 if(top==19)
printf("stack
else
stack[++top]=
}
int pop()
{
 int k;
if(top==-1)
return(0);
else
{
 k=stack[to
return(k);
}
}
```

Output :-

```
Enter the nu
Enter 1 If
Enter 1 If 2
Enter 1 If
Enter 1 If
The ADJACEN
0 1 1 1
0 0 0 1
0 0 0 0
0 0 1 0
```

```
for(i=1;i<=n;i++)
if(vis[i]==0)
dfs(i,n);
}
void push(int item)
{
if(top==19)
printf("stack overflows");
else
stack[++top]=item;
}
int pop()
{
int k;
if(top==-1)
return(0);
else
{
k=stack[top--];
return(k);
}
}
```

Output :-

```
Enter the Number of vertices 4
Enter 1 If 1 has a node with 1 else 0  0
Enter 1 If 2 has a node with 1 else 0  0
Enter 1 If 3 has a node with 1 else 0  0
Enter 1 If 4 has a node with 1 else 0  0
The ADJACENCY MATRIX IS
0 1 1 1
0 0 0 1
0 0 0 0
0 0 1 0
```

MENU
1. B.F.S
2. D.F.S
Enter your choice1
Enter the source vertex:1
1 2 3 4 Do you want to continue (Y/N) Y

Menu
1. B.F.S
2. D.F.S
Enter your choice2
Enter the source vertex:1
1 4 3 2 Do u want to CONTINUE (Y/N) N

15/6/23