PUNE INSTITUTE OF COMPUTER TECHNOLOGY
PUNE - 411043
Department of Electronics & Telecommunication
ASSESMENT YEAR: 2020-2021 CLASS: SE V
SUBJECT: Data Structure and Algorithm
Assg No: 6 Roll No: 22119 Date: 05/12/2020

Programmer Name: Param Chordiya
Batch: E5
Problem Statement:

Write a program to Implement Graph using adjacency Matrix, apply following traversal

1. Breadth First Search (BFS)
2. Depth First Search (DFS)

INPUT:

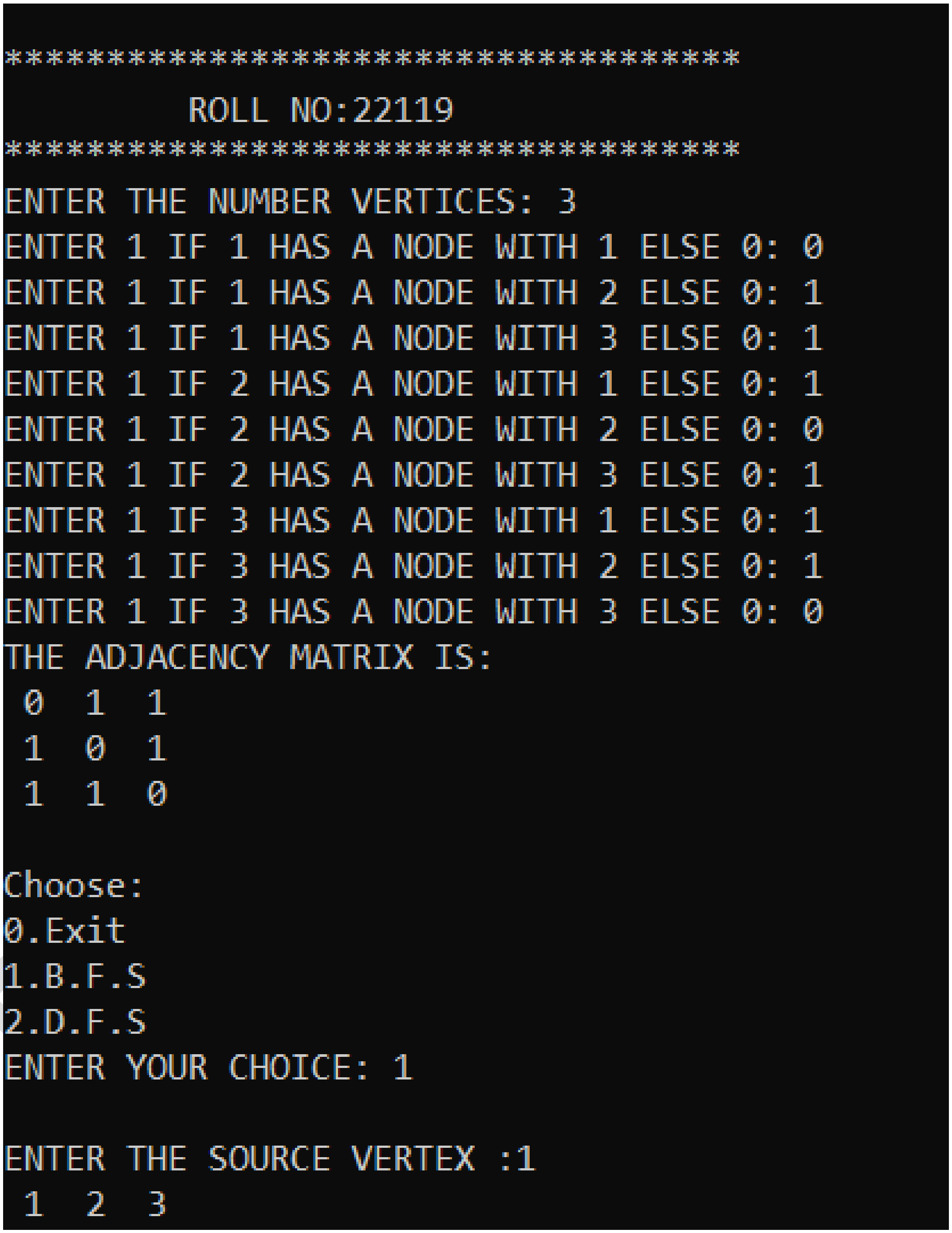
```
#include<stdio.h>
#include<stdlib.h>
int q[20],top=-1,front=-1,rear=-1,a[20][20],vis[20],stack[20];
int delete();
void add(int item);
void bfs(int s,int n);
void dfs(int s,int n);
void push(int item);
int pop();

void main()
{
                    printf("\n************************************");
                    printf("\n        ROLL NO:22119");
                    printf("\n************************************\n");
                    int n,i,s,ch,j;
                    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("\nChoose:\n0.Exit \n1.B.F.S \n2.D.F.S");
                printf("\nENTER YOUR CHOICE: ");
                scanf("%d",&ch);
                switch(ch){
                        case 1:printf("\nENTER THE SOURCE VERTEX :");
                                scanf("%d",&s);
                                bfs(s,n);
                                break;
                        case 2:printf("\nENTER THE SOURCE VERTEX :");
                                scanf("%d",&s);
                                dfs(s,n);
                                break;
                        default:printf("\nWrong choice!");
                }
            }while(ch!=0);
}

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);
            }
```

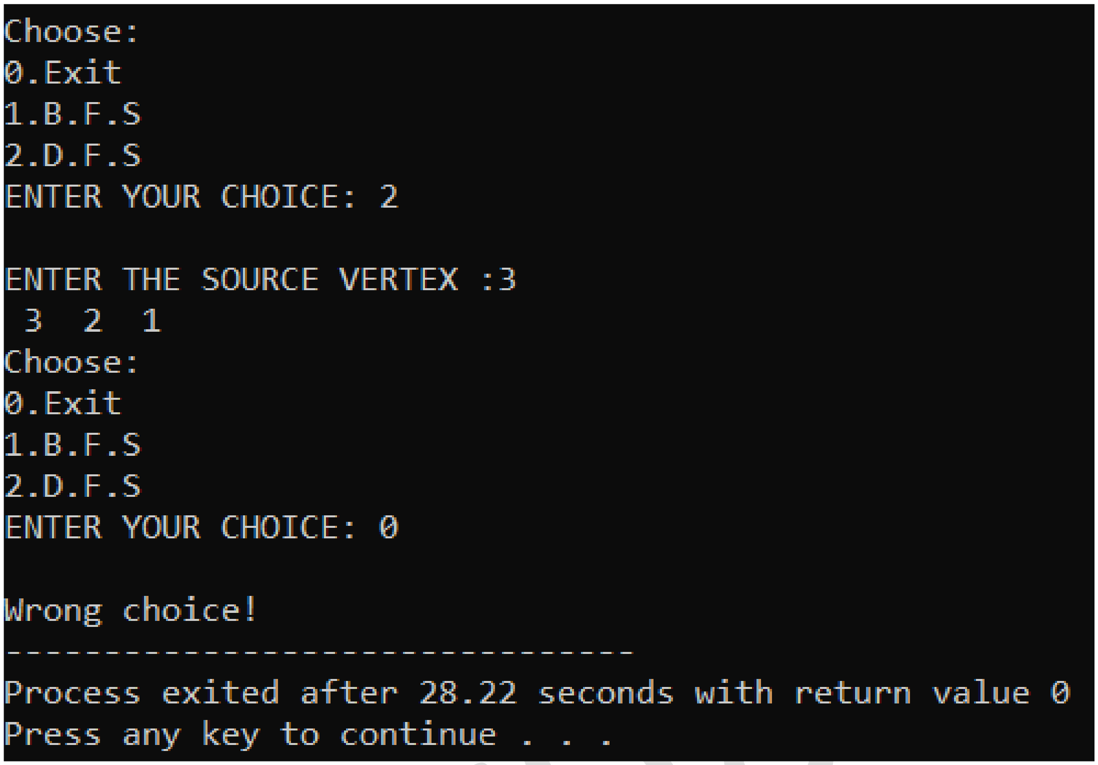
```
                        for(i=1;i<=n;i++)
                        if(vis[i]==0)
                        bfs(i,n);
}

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-1];
                                  return(k);
                              }
}
```

OUTPUT:



```
************************************
         ROLL NO:22119
************************************
ENTER THE NUMBER VERTICES: 3
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 2 HAS A NODE WITH 1 ELSE 0: 1
ENTER 1 IF 2 HAS A NODE WITH 2 ELSE 0: 0
ENTER 1 IF 2 HAS A NODE WITH 3 ELSE 0: 1
ENTER 1 IF 3 HAS A NODE WITH 1 ELSE 0: 1
ENTER 1 IF 3 HAS A NODE WITH 2 ELSE 0: 1
ENTER 1 IF 3 HAS A NODE WITH 3 ELSE 0: 0
THE ADJACENCY MATRIX IS:
 0  1  1
 1  0  1
 1  1  0

Choose:
0.Exit
1.B.F.S
2.D.F.S
ENTER YOUR CHOICE: 1

ENTER THE SOURCE VERTEX :1
 1  2  3
```



```
Choose:
0.Exit
1.B.F.S
2.D.F.S
ENTER YOUR CHOICE: 2

ENTER THE SOURCE VERTEX :3
 3  2  1
Choose:
0.Exit
1.B.F.S
2.D.F.S
ENTER YOUR CHOICE: 0

Wrong choice!
--------------------------------
Process exited after 28.22 seconds with return value 0
Press any key to continue . . .
```