

Write a C program to

1. DFS
2. BFS

12/6/17  
LAB - 2.

SPARK

Date \_\_\_\_\_ Page \_\_\_\_\_

```
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 push(int item);
int pop();

void main()
{
    int n, i, s, ch, j;
    char c, dummy;
    printf("Enter no. of vertices");
    scanf("%d", &n);
    for (i = 1; i <= n; i++)
    {
        for (j = 1; j <= n; j++)
            printf("Enter if %d has a node with
                   %d. If yes o", i, j);
        scanf("%d", &a[i][j]);
    }
    printf("Total The Adjacency matrix is %n");
    for (i = 1; i < n; i++)
    {
        for (j = 1; j <= n; j++)
            printf("%d", a[i][j]);
        printf("\n");
    }
}
```

y

printf ("In");

do

s

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

vis[i] = 0;

printf ("Mem");

printf ("I-BFS");

printf ("RDFS");

printf ("Inter choice");

Scans ("I.d", &ch);

printf ("Itu The source vertex");

Scans ("I.d", &s);

switch (ch)

s

case: bfs (s, n);

break;

case:

dfs (s, n)

break;

p

printf ("Do you want to continue (Y/N)?");

Scans ("I.C", &dummy);

Scans ("I.dr", &c);

y

while ((c == 'y' || (c == 'Y')));

j

void bfs (int s, int n)

{

int p, i;

add (s);

vis[s] = 1;

p = delete ();

if (p != 0)

printf ("Ld", p);

while (p != 0)

{

for (i = 1; i <= n; i++)

if ((adj[p][i]) != 0) && (vis[i] == 0))

add (i);

vis[i] = 1;

}

p = delete ();

if (p != 0)

printf ("Ld", p);

,

for (i = 1; i <= n; i++)

if (vis[i] == 0)

bfs (i, n);

}

"");

void add (int item)

{

if (rear == 19)

printf ("queue is full");

else

{

```
If [ rear == -1 )  
{  
    q[ ++ rear ] = item;  
    front++;
```

else

```
    q[ ++ rear ] = item;
```

y

int delete()

{

int k;

if ((front > rear) || (front == -1))

return (0);

else

{

K = q[ front++ ];

return (K);

y

A  
void dfs (int s, int u)

!

int i, k;

push(s);

vis[s] = 1;

K = pop();

if (K != 0)

print ("id" - K)

while (K != 0)

{

```
for (i = 1; i <= n; i + 1)
    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);
    }
}
```

Homework  
stack

10/10

```
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.bfs
2.dfs
enter choice1
enter source vertex :1
1 2 3 4 DO U WANT TO CONTINUE(Y/N) ? y
```

```
menu
1.bfs
2.dfs
enter choice2
enter source vertex :1
1 4 3 2 DO U WANT TO CONTINUE(Y/N) ?
```