**Program No. :- 8**

**AIM: Write a program to for the implementation of Breadth First Search Algorithm.**

**SOURCE CODE:**

#include<stdio.h>

#include<conio.h>

#include<limits.h>

#include<stdlib.h>

int edge[10][10],size,item;

char vertex[10];

int visited[10],distance[10];

int queue[10],head=0,tail=0,max=10,source,i,j,q;

void push(int item)

{

if(head==max-1)

{

printf("overflow");

}

else if(head==0)

{

head=1;

tail=1;

}

else

{

tail=tail+1;

}

queue[tail]=item;

}

int pop()

{

if(tail==0)

{

printf("underflow");

}

else if(head==tail)

{

item=queue[head];

head=0;

tail=0;

}

else

{

item=queue[head];

head=head+1;

}

return item;

}

void build\_graph()

{

printf("enter number of vertices");

scanf("%d",&size);

printf("enter %d vertices of graph",size);

for(i=0;i<size;i++)

{

fflush(stdin);

scanf("%c",&vertex[i]);

}

char ans;

printf("enter the adjancy matrix for the graph");

for(i=0;i<size;i++)

{

for(j=0;j<size;j++)

{

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

}

}

for(i=0;i<size;i++)

{

for(j=0;j<size;j++)

{

printf("%d\t",edge[i][j]);

}

printf("\n");

}

}

void bfs()

{

for(i=0;i<size;i++)

{

if(i!=source)

{

visited[i]=0;

distance[i]=INT\_MAX;

}

}

visited[source]=1;

distance[source]=0;

push(source);

while(head!=NULL)

{

q=pop();

for(i=0;i<size;i++)

{

if(edge[q][i]==1&&!visited[i])

{

visited[i]=1;

distance[i]=distance[q]+1;

push(i);

printf(" %c\t\t\t %d",vertex[i],distance[i]);

printf("\n");

}

}

}

}

void main()

{

clrscr();

build\_graph();

printf("enter source of graph ");

scanf("%d",&source);

printf("the vertices which are reachable from source are:\n\n vertex \t\t distance from source\n");

bfs();

getch();

}

**Output:**

