BFS (breadth first search)

#include<stdio.h>

#include<conio.h>

char que[20];

int front=0, rear=0, n;

char arr[20];

int bfs(int );

char ajMat[20][20];

char b[20];

void display();

int p=0;

int main()

{

char v;

printf("Enter the number of nodes in a graph");

scanf("%d",&n);

printf("Enter the value of node of graph");

for(int i=0; i<n; i++)

{

scanf("%s",&b[i]);

}

printf("Enter the value in adjancency matrix in from of 'y' or 'n'\n");

printf("If there exits an edge between two vertices than 'y' otherwise 'n'\n");

for(int i=0; i<n; i++)

printf(" %c ",b[i]);

for(int i=0;i<n; i++)

{

printf("\n%c ",b[i]);

for(int j=0; j<n; j++)

{

printf("%c ",v=getch());

ajMat[i][j]=v;

}

printf("\n\n");

}

for(int i=0;i<n;i++)

bfs(i);

display();

getch();

}

void display()

{

printf("BFS of Graph : ");

for(int i=0; i<n; i++)

printf("%c ",arr[i]);

}

void insert(char val)

{

que[front]=val;

front++;

}

char del()

{

rear=rear+1;

return que[rear-1];

}

bool unVisit(char val)

{

for(int i=0; i<front; i++)

{

if(val==que[i])

return false;

}

return true;

}

int bfs(int i)

{

char m;

if(front==0)

{

insert(b[i]);

}

for(int j=0; j<n; j++)

{

if(ajMat[i][j]=='y')

{

if(unVisit(b[j]))

{

insert(b[j]);

}

}

}

m=del();

arr[p]=m;

p++;

return 0;

}

DFS (Depth first search)

#include<stdio.h>

#include<conio.h>

char stack[20];

int top=-1, n;

char arr[20];

char dfs(int );

char ajMat[20][20];

char b[20];

void display();

int p=0;

int main()

{

char v;

int l=0;

printf("Enter the number of nodes in a graph");

scanf("%d",&n);

printf("Enter the value of node of graph");

for(int i=0; i<n; i++)

{

scanf("%s",&b[i]);

}

char k=b[0];

printf("Enter the value in adjancency matrix in from of 'Y' or 'N'\n");

printf("\nIf there is an edge between the two vertices then enter 'Y' or 'N'\n");

for(int i=0; i<n; i++)

printf(" %c ",b[i]);

for(int i=0;i<n; i++)

{

printf("\n%c ",b[i]);

for(int j=0; j<n; j++)

{

printf("%c ",v=getch());

ajMat[i][j]=v;

}

printf("\n\n");

}

for(int i=0;i<n;i++)

{

l=0;

while(k!=b[l])

l++;

k=dfs(l);

}

display();

getch();

}

void display()

{

printf(" DFS of Graph : ");

for(int i=0; i<n; i++)

printf("%c ",arr[i]);

}

void push(char val)

{

top=top+1;

stack[top]=val;

}

char pop()

{

return stack[top];

}

bool unVisit(char val)

{

for(int i=0; i<p; i++)

if(val==arr[i])

return false;

for(int i=0; i<=top; i++)

if(val==stack[top])

return false;

return true;

}

char dfs(int i)

{

int k;

char m;

if(top==-1)

{

push(b[i]);

}

m=pop();

top--;

arr[p]=m;

p++;

for(int j=0; j<n; j++)

{

if(ajMat[i][j]=='y')

{

if(unVisit(b[j]))

{

push(b[j]);

}

}

}

return stack[top];

}