DATA STRUCTURE AND ALGORITHM

Exercise 10 Breadth First Search and Depth First Search

Name: A H K Swarun

Reg no: 21BPS1252

Q. Consider the below graph and perform the graph Traversals (BFS and DFS)

Algorithm:

```
Algorithm bfs&dfs( adj[][MAX],visited[],start, i, j){ breadthfirstsearch(){
queue[MAX],rear=-1,front=-1,i,k;
for(k=0;k<MAX;k++)
   visited[k]=0;
   queue[++rear]=start;
   ++front; visited[start]=1;
while(rear>=front){
   start=queue[front++]; printf(,start
   + 65);
for(i=0;i<MAX;i++){
   if(adj[start][i] && visited[i] == 0)
   {
      queue[++rear]=i;
      visited[i]=1;
   }
}
}
}
```

depthfirstsearch(adj[][MAX],visited[],start){

stack[MAX],top=-1,i,k;

for(k=0;k<MAX;k++)

{visited[k]=0;

stack[++top]=start;

visited[start]=1;

while(top != -1){

start=stack[top--]; printf(start + 65);

for(i=0;i<MAX;i++){

if(adj[start][i] && visited[i] == 0)

{

stack[++top]=i; visited[i]=1;

break;

}

}

}

}

}

Code:

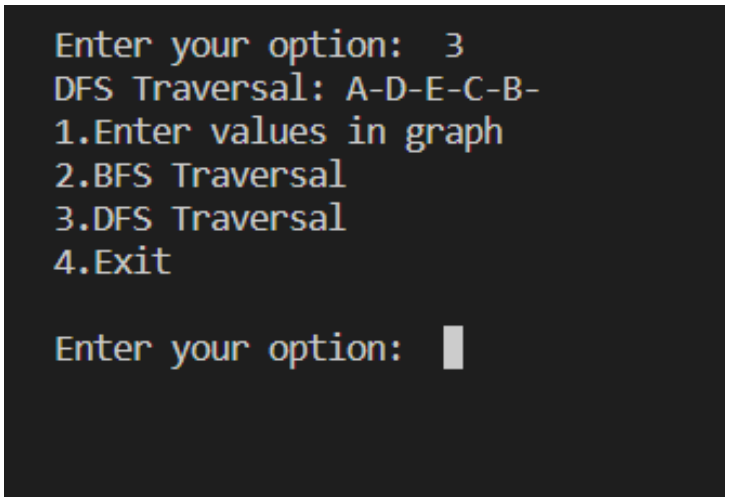
```
#include<stdio.h>
#include<conio.h>
#define MAX 10

void breadth_first_search(int adj[][MAX],int visited[],int start){
     int queue[MAX],rear=-1,front=-1,i,k;
     for(k=0;k<MAX;k++) visited[k]=0;
     queue[++rear]=start;
     ++front;
     visited[start]=1;
    while(rear>=front){
        start=queue[front++]; printf("%c-""\n",start + 65);
        for(i=0;i<MAX;i++){
            if(adj[start][i] && visited[i] == 0){
                queue[++rear]=i;
                visited[i]=1;
```

```
            }
        }
    }
}

void depth_first_search(int adj[][MAX],int visited[],int start){
     int stack[MAX],top=-1,i,k;
     for(k=0;k<MAX;k++) visited[k]=0;
     stack[++top]=start; visited[start]=1; while(top != -1){
        start=stack[top--]; printf("%c-""\n",start + 65);
        for(i=0;i<MAX;i++){
            if(adj[start][i] && visited[i] == 0){
                stack[++top]=i;
                visited[i]=1;
            }
        }
    }
}

int main(){
    int visited[MAX]={0}; int adj[MAX][MAX],i,j; int option,size;
    do{
        printf("\n1.Enter values in graph");
        printf("\n2.BFS Traversal");
        printf("\n3.DFS Traversal");
        printf("\n4.Exit");
        printf("\n\nEnter your option: ");
        scanf("%d",&option);
        switch(option){
            case 1:
            printf("\nEnter the adjaceny matrix:\n");
            for(i=0;i<MAX;i++){
                for(j=0;j<MAX;j++){
                scanf("%d",&adj[i][j]);
                }
            }
            break;
            case 2:
            printf("BFS Traversal:");
            breadth_first_search(adj,visited,0);
            break;
            case 3:
            printf("DFS Traversal:");
            depth_first_search(adj,visited,0);
            break;
```

```
        }
    }
    while(option!=4);
    getch();
    return 0;
}
```

Output:

```
PS C:\Users\ahks4> cd Desktop
PS C:\Users\ahks4\Desktop> gcc graphs.c
PS C:\Users\ahks4\Desktop> .\a.exe

1.Enter values in graph
2.BFS Traversal
3.DFS Traversal
4.Exit

Enter your option:  1

Enter the adjaceny matrix:
0 1 1 1 0
0 0 0 1 0
0 1 0 0 0
0 0 0 0 1
0 0 1 0 0

1.Enter values in graph
2.BFS Traversal
3.DFS Traversal
4.Exit

Enter your option:  2
BFS Traversal: A-B-C-D-E-
1.Enter values in graph
2.BFS Traversal
3.DFS Traversal
4.Exit
```


Enter your option: 3
DFS Traversal: A-D-E-C-B-
1.Enter values in graph
2.BFS Traversal
3.DFS Traversal
4.Exit
Enter your option: