

```

1  #include<stdio.h>
2  #include<stdlib.h>
3  #define MAX_VERTICES 100
4
5  struct Node
6  {
7      int data;
8      struct Node *next;
9  };
10
11 struct Node *createNode(int data)
12 {
13     struct Node *newnode=(struct Node *)malloc(sizeof(struct Node));
14     newnode->data=data;
15     newnode->next=NULL;
16     return newnode;
17 };
18
19 void addEdge(struct Node *adjlist[],int u,int v)
20 {
21     struct Node*newnode=createNode(v);
22     newnode->next=adjlist[u];
23     adjlist[u]=newnode;
24 }
25
26 void BFS(struct Node *adjlist[],int vertices,int startNode,int visited[])
27 {
28     int queue[MAX_VERTICES];
29     int front=0,rear=0;
30     visited[startNode]=1;
31     queue[rear++]=startNode;
32     while(front<rear){
33         int currentNode=queue[front++];
34         printf("%d\t",currentNode);
35         struct Node *temp=adjlist[currentNode];
36         while(temp!=NULL){
37             int neighbour=temp->data;
38             if(!visited[neighbour]){
39                 visited[neighbour]=1;
40                 queue[rear++]=neighbour;
41             }
42             temp=temp->next;
43         }
44     }
45 }
46
47 int main()
48 {
49     int vertices=5,i;
50     struct Node *adjlist[vertices];
51     for(i=0;i<vertices;i++)
52         adjlist[i]=NULL;
53     addEdge(adjlist,0,1);
54     addEdge(adjlist,0,2);
55     addEdge(adjlist,1,3);
56     addEdge(adjlist,1,4);
57     addEdge(adjlist,2,4);
58     int visited[vertices];
59     for(i=0;i<vertices;i++)
60         visited[i]=0;
61     printf("BFS start vertex ");
62     BFS(adjlist,vertices,0,visited);
63     return 0;
64 }

```