```
/* create a graph using list representation/*
#include<stdio.h>
#include<stdlib.h>
struct node{
char vertex;
struct node *next;
};
void create(struct node *adj[],int n);
void display(struct node *adj[],int n);
void delete(struct node *adj[],int n);
int main(){
        struct node *adj[10];
        int i,n;
        printf("\n Enter no of nodes :");
        scanf("%d",&n);
                for(i=0;i<n;i++){
                   adj[i]=NULL;   }
         create(adj,n);
         printf("\n Graph is crated :\n");
         display(adj,n);
return 0;
}
void create(struct node *adj[],int n){
  struct node *new_node,*last;
        int i,j,no_adj_node;
        int val;
                for(i=0;i<n;i++){
                        last=NULL;
                                printf("\n Enter no of adjacent node of node %d : ",i);
```

```
                    scanf("%d",&no_adj_node);
                    for(j=0;j<no_adj_node;j++){
                        printf("\n Enter a value of a node :");
                        scanf("%c",&val);
                        new_node=(struct node *)malloc(sizeof(struct node));
                        new_node->vertex=val;
                        new_node->next=NULL;
                            if(adj[i]==NULL)
                                adj[i]=new_node;
                            else
                                last->next=new_node;
                            last=new_node;
                    }
                }
}
void display(struct node *adj[],int n){
        struct node *ptr;
        int i;
           for(i=0;i<n;i++){
              ptr=adj[i];
              printf("\n The neighbours of node %c ", ptr->vertex);
              while(ptr!=NULL){
                    printf("\n\t %d",ptr->vertex);
                    ptr=ptr->next;
              }
           }
}
```