Name: Vishal Bhimgonda Desai

Roll no: B78

PRN : 2425010086

C Program to Implement BFS

```
#include <stdio.h>
#include <stdlib.h>

#define MAX_VERTICES 100

struct Node {
   int data;
   struct Node* next;
};

struct Node* createNode(int data) {
   struct Node* newNode = (struct Node*)malloc(sizeof(struct Node));
   newNode->data = data;
   newNode->next = NULL;
   return newNode;
}

void addEdge(struct Node* adj[], int u, int v) {
   struct Node* newNode = createNode(v);
   newNode->next = adj[u];
   adj[u] = newNode;

   newNode = createNode(v);
   newNode->next = adj[u];
   adj[u] = newNode;
```

```
}

void bfs(struct Node* adj[], int V, int s, int visited[]) {
  int queue[MAX_VERTICES];
  int front = 0, rear = 0;

  visited[s] = 1;
  queue[rear++] = s;

  while (front != rear) {
    int curr = queue[front++];
    printf("%d ", curr);

    struct Node* temp = adj[curr];
    while (temp != NULL) {
      int neighbor = temp->data;
      if (!visited[neighbor]) {
        visited[neighbor] = 1;
        queue[rear++] = neighbor;
      }
      temp = temp->next;
    }
  }
}

void bfsDisconnected(struct Node* adj[], int V) {
  int visited[V];
  for (int i = 0; i < V; i++) {
    visited[i] = 0;
  }
```

```
    for (int i = 0; i < V; ++i) {
        if (!visited[i]) {
            bfs(adj, V, i, visited);
        }
    }
}

int main() {
    int V = 6;
    struct Node* adj[V];

    for (int i = 0; i < V; ++i) {
        adj[i] = NULL;
    }

    addEdge(adj, 0, 1);
    addEdge(adj, 0, 2);
    addEdge(adj, 3, 4);
    addEdge(adj, 4, 5);

    bfsDisconnected(adj, V);

    return 0;
}
```

Output:

0 1 2 3 4 5