```
#include <iostream>
using namespace std;
struct Node{
   string data;
   Node* next;
};

class Route{
   Node* tail;        //declare a tail pointer
   public:
      Route(){
         tail = NULL;    //initializing tail to null
      }

 //function to add new route i.e. to add new node in linked list
      void add_route(string value){
          //declare a new node to add in list
           Node* nn = new Node;
         //assning value to data of new node
           nn->data = value;

 //check if there is no previous node then our tail and tail->next should be same
         if(tail == NULL){
            nn->next = nn;
            tail = nn;
         }
         //if prevoius nodes are there in the list
         else{
            nn->next = tail->next;
            tail->next = nn;
            tail = nn;
         }
      }

      //Function for deleting routes
      void del(string value){

         //if there is no route to delete then print the message
         if(tail == NULL){
            cout<<"There are no routes to delete"<<endl;
            return;
         }

         //if there is only one node and it contains the value
```

```
        if(tail == tail->next && tail->data == value){
            delete tail;       //deletes the value at that node
            tail = NULL;       //address of tail also set null
            return;
        }

        //to keep track the value is found or not
        int flag = 0;

        //if we delete the value at current node then prevous nodes next sholud points to the
next node by skipping the current node and in list we are not allowed to go back so we need two
pointers
        Node* temp = tail->next;
        Node* p = tail;

        do{
            //check if value to be deleted is matched with value at that node
            if(temp->data == value){
                flag = 1;

                //move previous pointer by skipping the current node to next node
                p->next = temp->next;

                //store the value to be deleted in temporary variable to free the memory
                Node* ntbd = temp;
                temp = temp->next;
                delete ntbd;      ///delete the current node
            }

            else{
                p = temp;   //move p pointer to temp
                temp = temp->next; //move temp pointer to next node;
            }

        }while(temp != tail->next); //enter the loop till we traverse all the nodes

        //if value not found then print message
        if(flag == 0){
        cout<<"Value to be deleted not found"<<endl<<endl;
        }
    }
    void print(){
        // if there is no node then printing printing that message
```

```
        if(tail == NULL){
            cout<<"Route is not available for delivering a product"<<endl;
            return;
        }

        //if there are 1 or more nodes then declare a temporary pinter to traverse through a
linked list

        Node* temp = tail->next;
        cout<<"Available routes : "<<endl;

        //using do while beacause at first node and last node value of tail->next is same in
circular linked list

        do{
          cout<<temp->data<<" to "<<temp->next->data<<endl;
          temp = temp->next;       //move to next node
        }while(tail->next != temp);
      }
};

int main() {
    Route r;
    r.add_route("Pune");
    r.add_route("Mumbai");
    r.add_route("Delhi");
    r.add_route("Nashik");
    r.add_route("Kolhapur");
    r.print();
    cout<<endl;
    r.del("Mumbai");
    r.print();
    cout<<endl;
    r.del("trf");
    r.print();
    return 0;
}
```